National Research Council Canada
Institute for Information Technoiogy
Conseil national de recherches Canada
Institut de Technologie de l'information
ERB-1068

NRC·CNRC

Evaluating Capture-recapture Models with Two Inspectors

Khaled El-Emam and Oliver Laitenberger
December 1999

Canada
NRC 43613

National Research Council Canada
Institute for Information Technology

Conseil national de recherches Canada
Institut de Technologie de l'information

Evaluating Capture-recapture Models with Two Inspectors

Khaled El-Emam and Oliver Laitenberger
Decmeber 1999

Copyright 1999 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

102-05436/

Evaluating Capture-Recapture Models with Two Inspectors

Khaled El Emam
National Research Council, Canada
Institute for Information Technology
Building M-50, Montreal Road
Ottawa, Ontario
Canada K1A OR6
Khaled.El-Emam@iit.nrc.ca

Oliver Laitenberger
Fraunhofer Institute for
Experimental Software Engineering
Sauerwiesen 6
D-67661 Kaiserslautern
Germany
+49 (0)6301 707251
laiten@iese.fhg.de

Abstract

Capture-recapture (CR) models have been proposed as an objective method for controlling software inspections. CR models were originally developed to estimate the size of animal populations. They have also been used to estimate the number of defects in an inspected artifact. Armed with this estimate, one can decide whether the artifact requires a reinspection to ensure that a minimal inspection effectiveness level has been attained. Little evaluative research has been performed thus far on the utility of CR models for inspections with two inspectors. Furthermore, these studies have focused on the relative error of the defect content estimates exclusively. In this paper we report on an extensive Monte Carlo simulation that evaluated six capture-recapture models for two inspectors assuming a code inspections context. In addition to relative error, we evaluate the accuracy of the reinspection decision. The latter is more congruent with the manner in which these models would be used in practice. Our results indicate that the most appropriate capture-recapture model for two inspectors is an estimator originally developed by Chapman that allows for inspectors with different capabilities. This will have a relatively high decision accuracy and will perform better than the default decision of no reinspections. Furthermore, we identify the conditions under which this estimator will perform best.

1 Introduction

A recent literature review found that, *on average*, software inspections find only 57% of defects in code and design documents [8]. Given the substantial defect detection cost savings that can be accrued by increasing the effectiveness[1] of inspections [8], contemporary research has focused on improved reading techniques (e.g., see [33][3][19][41]) and on reinspections (e.g., see [24]) for maximizing effectiveness. The focus of this paper is on maximizing inspection effectiveness through reinspections.

Reinspections can be considered part of the general problem of when to stop inspections. As is the case with testing, one needs a criterion by which to decide whether a document should be inspected anew, or whether it can pass to the subsequent phase.

Most organizations have not institutionalized procedures for deciding when to stop software inspections. Those that do have utilized, for example, historical norms so that if too many defects are found compared to the norm then this is taken as evidence of a poor document, while too few are taken as evidence of a poor inspection [24]. However, this approach assumes that variations among reviews are larger than variations among documents. If this is not the case then this can lead to reinspections of high quality documents, and low quality documents may easily pass.

To address these potential problems, one can use Capture-Recapture (CR) models. CR models were initially developed to estimate the size of animal populations (e.g., see [38][51]). In a software engineering context, they have been applied in controlling the testing process [4][30][36][21][37], and more recently they have been used in controlling the inspection process [23][24].

[1] Effectiveness is defined as the proportion of defects in a document that were found during the inspection.

When applied to software inspections, CR models can be used to estimate the number of defects in the inspected document. Using this estimate and the known number of defects found, the number of remaining defects in the inspected document can be estimated. Subsequently, armed with this information, the inspection team can make the decision as to whether the document should be reinspected to reduce its defect content before passing it on to the next phase of the life cycle.

Researchers at Bell Labs first applied CR models for requirements and design inspections [23][24][25]. However, in these studies the true number of defects was unknown and therefore an evaluation of their true efficacy was not possible. Later work consisted of a Monte Carlo simulation to evaluate the robustness of different CR estimators to violations of their assumptions [50].

Objective empirical evaluation of CR models started with the study of Wohlin et al. [53]. However, this study was conducted with non-software engineering documents. Subsequent work used software engineering artifacts [10][12][35][44]. All of the above work utilized models that were originally developed in wildlife research. Other researchers considered the incorporation of Bayesian methods to estimate defect content [5], performed further evaluations of assumption violations when using CR estimates [48], and evaluated the applicability of CR models to perspective-based reading [12][49].

An alternative approach was proposed in [54], the Detection Profile Method (DPM). The DPM is an intuitively appealing approach that can be easily explained graphically to nonspecialists. A later study suggested a method for selecting between a CR model and the DPM [9], and this was subsequently further evaluated in [39].

In addition to the experiences reported by the researchers at Bell Labs, the use of the DPM at an insurance company in Germany was reported in [11], and the application of CR models in telecommunications projects [1]. Therefore, there is a growing adoption of defect content estimation models in industrial practice, and specifically CR models.

Little empirical investigation of the utility of CR models for inspections with two inspectors has been conducted. Furthermore, these studies have exclusively evaluated the accuracy with which CR models can predict the number of defects in an artifact or the accuracy of the estimated number of remaining defects. However, given that the objective is to make a *reinspection decision*, it is also necessary to evaluate the *decision accuracy* using a CR model.

In this paper we present the results of an extensive Monte Carlo simulation that evaluates the relative error, dispersion, failure rate, and decision accuracy of biological CR models[2] for two-person inspections. The advantage of a simulation is that we can obtain a general picture of the utility of CR models with two inspectors. The simulation defined 48 study points that varied defect difficulty, inspector capability, and the proportion of difficult defects in an inspected artifact. The objectives of the simulation were twofold:

- Identify the best performing CR model in terms of decision accuracy
- Identify the impact of assumption violation on the decision accuracy of the different CR models.

To our knowledge, this is the first comprehensive Monte Carlo evaluation of all biological CR models for two person inspections. Furthermore, it is the first study that explicitly evaluates the reinspection decision accuracy.

Our results indicate that the best CR model for making the reinspection decision is one that was originally devised by Chapman. This estimator allows for inspectors with different capabilities and has low failure rates. Furthermore, it has a relatively high decision accuracy and performs better than the default decision of always passing a document to the next phase without a reinspection. We also identified the conditions under which this estimator would work best.

The paper is organized as follows. Section 2 presents background information about two-person inspections and reinspections in practice. In Section 3 we provide an overview of CR models. In Section 4 we specify our simulation parameters, and describe how the results were evaluated. Our results are

---

[2] We do not consider other approaches such as the DPM and its extensions here since our concern in this study is capture-recapture models with biological origins. Furthermore, the DPM as it is defined would be difficult to apply with data from only two inspectors.

presented in Section 5. We conclude the paper with a summary and directions for future work in Section 6.

2 Background

This section provides the background for our study in terms of reinspections and two-person inspections. We first illustrate that performing reinspections is not a common practice in software engineering. Therefore, the default practice is to pass all documents after fixing defects found during a single inspection. Later in this paper we evaluate whether using CR models is better than such common practice. Following that we review the evidence illustrating that performing inspections with two inspectors, as opposed to a single or greater than two inspectors, can be cost effective. This indicates that two-person inspections are indeed an effective inspection team size.

2.1 Reinspections in Software Engineering Practice

A reinspection, as used in this paper, is intended to scrutinize an inspected document anew. The purpose is to identify defects that have been missed during the initial inspection. It is not to focus on the changes made due to the initial inspection.

Some inspection implementations involve a follow-up phase at the end of the inspection process. Fagan [27] reports that this inspection phase aims at verifying whether the author has taken some remedial action for each issue, problem, and concern detected. He also states that the follow-up phase is an optional one in the inspection process and that it cannot be considered a reinspection.

In their book on software inspections, Strauss and Ebenau [47] describe the reinspection stage. However, the focus of this is to concentrate on the changes made after the initial inspection, their interfaces and dependencies. This is different from performing a reinspection to identify defects that have been missed.

It seems, however, that some companies or projects regard the follow-up phase as a way to mitigate the risk of remaining defects without spending the effort on a reinspection. Shirey, for example, presents some results from a survey at Hewlett-Packard [46]. He found that over half of those questioned had never reinspected anything. But even if the collected inspection data indicated a need for a reinspection, it is not performed in each and every case. Barnard and Price [2] present inspection results from AT&T in which modules were not reinspected although the inspection data recommended it. They state that this finding needs further investigation but provide no additional explanation.

Therefore, it is reasonable to conclude that in practice few organizations have institutionalized procedures for deciding when to stop inspecting. Even when reinspections are clearly warranted, this is not systematically performed.

2.2 Two-Inspector Inspections

In this paper, we focus on two inspector inspections. This means that two persons independently scrutinized the software artifact for defects before the inspection meeting takes place. This does not necessarily imply a limit to the overall inspection team size since other people may be involved in the inspection process, such as an independent moderator. However, the quality of the inspection process as well as the quality of the artifact after inspection is primarily determined by those people who scrutinize the artifact for defects (i.e., the inspectors).

Involving only two inspectors is in line with suggestions in Fagan's original work on software inspection [27]. He states that four people (i.e., the inspection moderator, the author, and two inspectors) constitute a good-sized inspection team, although circumstances may dictate otherwise. Such circumstances may be, for example, that a requirements document is inspected instead of a code artifact. Since the requirements cannot be checked against a preceding specification, a requirements inspection often involves more inspectors than other inspection types [28]. For code inspections, however, empirical evidence suggests that adding inspectors does not necessarily pay-off in terms of more detected defects. In a controlled experiment, Porter et al. [42] investigated 1, 2, and 4 inspector inspections. They found little difference in the inspection effectiveness of 2 and 4 inspectors. However, both were significantly

more effective than 1 inspector inspections. There is also some evidence from academic environments that limiting the overall inspection team size to two people is an effective approach. In this case one inspector and the author, who also acts as an inspector. It was found that such a team constellation decreases inspection cost while maintaining inspection effectiveness [7][31].

Based on the suggestions and the available empirical findings in the inspection literature, we can conclude that a two inspector approach represents an appropriate number of inspectors.[3]

3 Overview of CR Models

In biology, capture-recapture studies are used to estimate the size of an animal population. In doing so, animals are captured, marked, and then released on several trapping occasions. The number of marked animals that are recaptured allows one to estimate the total population size based on the samples' overlap. When many marked animals are recaptured, one can argue that the total population size is small and vice versa.

The capture-recapture principle in biology can be transferred to inspections: each inspector draws a sample from the population of defects in the inspected software artifact. In this way, an inspector is equivalent to a particular trapping occasion in biology. A defect discovered by one inspector and rediscovered by another is said to be recaptured. Based on estimators similar to the ones used in biology, the total number of defects in the software artifact can be estimated.

The basic idea behind a CR model can be illustrated with reference to Table 1 (see [52]). Here we have the defects found by both inspectors. The value n_{11} is the number of defects found by both inspectors. The values in parantheses are unknown. Therefore we do not know n_{22} which is the number of defects not found by either of the inspectors.

		Found by Inspector 2		
		Yes	**No**	
Found by Inspector 1	**Yes**	n_{11}	n_{12}	n_{1+}
	No	n_{21}	(n_{22})	(n_{2+})
		n_{+1}	(n_{+2})	(N)

Table 1: Incomplete contingency table with observed values of defects found from two inspectors.

The odds ratio can be estimated by:

$$\hat{\alpha} = \frac{n_{11}n_{22}}{n_{12}n_{21}} \qquad \textbf{Eqn. 1}$$

Under independence, the odds ratio has a value of 1. Therefore, by rearranging Eqn. 1, we can obtain an estimate of n_{22}:

$$\hat{n}_{22} = \frac{n_{12}n_{21}}{n_{11}} \qquad \textbf{Eqn. 2}$$

The total number of defects in the document can be obtained by:

$$\hat{N} = \frac{n_{12}n_{21}}{n_{11}} + n_{11} + n_{12} + n_{21} = \frac{n_{1+}n_{+1}}{n_{11}} \qquad \textbf{Eqn. 3}$$

[3] A similar conclusion was also drawn by Glass based on his overview of the literature [29].

Eqn. 3 is known as the Lincoln-Petersen estimate, and is commonly used in practice to estimate the size of animal populations. This estimator makes the assumption that all defects have the same probability of being detected. There are other types of models that relax this assumption, and/or invoke further assumptions. These are reviewed below.

3.1 Types of CR Models

The different types of CR models that have been proposed in a biological context make different assumptions about capture probability (see Pollock [40] and Seber [45] for overviews). This means the probability of an animal being caught and the probability of catching an animal in a specific trapping occasion.

The first type of models assume that there is a *time response*. In biology, it models the fact that on different days the capture probabilities of animals might vary. For example, small mammals tend to stay in their dry homes during rainy weather. Therefore, the probability of capturing a small mammal is higher for days with fine weather than for days with rainy weather. For inspections this can be used to model inspectors with different abilities to detect defects. For example, experienced inspectors find more defects than inexperienced inspectors and therefore have a higher probability of detecting defects.

The second type of models assume that there is *heterogeneity*. In biology, it models the fact that different animals vary in their capture probability. For example, older animals are less mobile than younger ones and stay more often in their homes. Therefore, the probability of capturing an old animal is smaller than of capturing a young animal. For inspections this can be used to model defects that differ in their detection probability. For example, defects that are hard to detect have a lower detection probability than defects that are easy to detect.

The above two types of models can account for the fact that defect detection probability can be affected by both inspectors and defects. Inspectors may have different detection capabilities due to variation in their ability to detect defects (due to experience, education, or reading technique used) and defects may have different detection probabilities when there are defects that are easier to detect than others.

Four capture-recapture models can therefore be formulated. Model M0 assumes that none of these sources of variation exist, Model Mt and Mh account for exactly one source of variation, and Model Mth accounts for both sources of variation. When the analogy is made to inspections, these models make the following assumptions about inspectors and defects:

1. Model M0 - No variation: All different defects have the same detection probability, and all inspectors have the same detection capability.
2. Model Mh - Variation by heterogeneity: Different defects can vary in their detection probability, but all inspectors have the same detection capability.
3. Model Mt - Variation by time response: All different defects have the same detection probability, but the inspectors have different detection capabilities. Hence, with this source of variation accounted for, a model allows for inspectors with differing "general ability". Note that this "general ability" affects all defects.
4. Model Mth - Two sources of variation are combined: time response and heterogeneity. This allows for different detection probabilities for the different defects and inspectors.

In addition to these sources of variation, Otis et. al. [38] and White et. al [51] consider variations due to *behavioral* or *trap response*. This reflects the fact that an animal may change its behavior due to the process of being captured and marked. For example, when using baited traps, the probability to get caught for the first time is less than the probability for subsequent captures. For instance, this is because animals can get fascinated by traps, so marked animals are more likely to get caught than unmarked animals. In inspections, this may be usable to model the fact that defects captured by more than one inspector have usually a higher probability of being detected. However, the estimators for this source of variation depend on the order of trapping occasions (i.e., inspectors). Since no ordering of inspectors seems reasonable in the context of inspections, this type of model is not considered adequate for a software engineering context. Table 2 summarizes the models considered here:

Model	Source(s) of Variation
M0	Defects are equal with respect to their probability of being detected. The probability of detecting defects among inspectors is the same.
Mt	Defects are equal with respect to their probability of being detected. The probability of detecting defects among inspectors varies.
Mh	Defects have different probabilities of being detected. The probability of detecting defects among inspectors is the same.
Mth	Defects have different probabilities of being detected. The probability of detecting defects among inspectors varies.

Table 2: Assumptions of the Capture-Recapture models.

When applying capture-recapture models for estimating the number of defects, suitable estimators are necessary. While the model defines the assumptions made about detection probabilities, the corresponding estimator is a formula that actually performs the estimation based on the model's assumptions. In order to derive these estimators, the models' assumptions have to be cast into a stochastic form. Using estimation techniques, such as maximum likelihood estimation (MLE), estimators can be derived. For one model several estimators can be derived by applying different estimation techniques. Table 3 summarizes the estimators that have been considered in software engineering for each type of model.

Model	Estimator	Notation
M0	Maximum Likelihood Estimator [38]	M0
Mt	Maximum Likelihood Estimator [38], Chao's Estimator [17]	MtMLE[4] MtCh[5]
Mh	Jackknife Estimator [13] Chao's Estimator [15][16]	MhJE[6] MhCh
Mth	Chao's Estimator [18]	MthCh

Table 3: Relevant capture-recapture models and considered estimators. [7]

In our study we consider all of the above models.[8] For ease of presentation, in this paper we will refer to them by the notation in Table 3 (e.g., the MtMLE model rather than the Mt model with the ML estimator).

3.2 Evaluation of Capture-Recapture Models

At the outset, we define the following notation:

---

[4] For two inspectors this is the same as the well known Lincoln-Peterson estimator shown earlier.

[5] For two inspectors, this estimator is the same as the one proposed by Chapman [14]. He noted that with small sample sizes the traditional Lincoln-Peterson estimator can be biased, and therefore his estimator corrects for such biases.

[6] In our study we used the testing method in [13] for selecting an appropriate order of the jackknife estimator up to the fifth order.

[7] See [12] for a description of the data that would need to be collected during an inspection for each of these types of models and estimators.

[8] It should be noted that, to our knowledge, ours is the first study that evaluates model MthCh for two inspectors. The reason being that Chao provided three estimators under this model. Only the third estimator is implemented in the software that has been most commonly used in previous studies (the software is described in [43]). The third estimator, by definition, always fails with two inspectors because it encounters a divide by zero. For our study we use the second estimator that has been provided by Chao.

$\hat{N}$	The estimate of the number of defects in the document
N	The actual number of defects in the document.
n_i	The number of defects found by inspector i, where i=1,2
D	The number of unique defects found by the inspection team.
f_i	The number of defects found i times, where i=1,2

To date, all previous empirical evaluations of CR models (and the DPM) have compared the predicted number of defects in an artifact with the actual number of defects (or variations such as the number of estimated remaining defects with the number of actual remaining defects), for example see [10][9][48][39][49][44][50]. To be specific, many articles use the relative error defined as follows:

$$RE = \frac{\hat{N} - N}{N} \qquad \textbf{Eqn. 4}$$

Using relative error to evaluate CR models is useful for understanding the behavior of the CR models, especially the extent of over/under- estimation in software engineering contexts. However, relative error is not sufficient for evaluating how CR models will perform in practice. This is elaborated upon below.

Relative error is *not congruent* with the manner in which it has been suggested that CR models be used in an inspections context. For example, it has been stated that "The [capture-recapture] method is based on the review information from the individual reviewers and through statistical inference, conclusions are drawn about the remaining number of defects after the review. This would allow us to take informed and objective decisions regarding whether to continue, do rework, or review some more." [54][53], and "One approach to optimize the effectiveness of inspections is to reinspect an artifact that is presumed to still have high defect content. The reinspection decision criterion could be based on the number of remaining defects after an inspection, which can be estimated with defect content models." [10]. Therefore, the current literature describes a *binary decision* being made using the estimates: pass or reinspect. By using the relative error, one is actually imposing harder requirements on the performance of CR models. This is illustrated below.

Let us say that an inspection was performed on a document with 30 defects, and that the inspection found 20 of these. Therefore the inspection effectiveness is 0.66. Also, let the effectiveness threshold imposed by the organization be 0.57. This means that the organization wants to ensure that its inspections attain at least 57% effectiveness. We have a CR model that underestimates systematically by 20%. In this case, the model would estimate that the document has 24 defects, giving an estimated effectiveness of 0.833. The decision based on the model's estimate would be to pass the document to the next phase since the inspection attained the minimal effectiveness. Therefore, even though the CR model exhibits underestimation of 20%, it still gives the correct decision.

As another example, we consider the case of extreme outliers. Some of the CR evaluation literature has shown a concern with extreme outliers [9]. The concern was based on the argument that if a model exhibits extreme outliers then inspectors using that model will have a diluted confidence in all of its estimates. Let us say that the CR model has extreme overestimates, say 300%. The estimated effectiveness when only 10 defects out of 30 are found is 0.11. The actual effectiveness is 0.33. For an effectiveness threshold of 0.57, the model that exhibits extreme overestimation still gives the correct decision: reinspect.

The above exposition makes clear that evaluating the relative error of a CR model is insufficient to inform us about the reinspection decision accuracy that one would expect in practice. It is therefore also necessary to evaluate the decision accuracy of CR models directly.

3.3 Objective of Our Simulation

The objectives of our simulation were twofold:

- Identify the best performing CR model in terms of decision accuracy
- Identify the impact of assumption violation on the decision accuracy of the different CR models.

To our knowledge, this is the first comprehensive Monte Carlo evaluation of all biological CR models for two person inspections. Furthermore, it is the first study that explicitly evaluates the reinspection decision accuracy.

Thus far, there has been only one published empirical study that evaluated the performance (in terms of relative error) of CR models with two inspectors using a data set from an experiment [10]. Therefore, for two person inspections this previous study does not inform us about the *general* utility of CR models for two inspectors. The authors concluded that none of the biological models that were studied were applicable with two inspectors. Furthermore, model Mth was not evaluated.

3.4 Previous Simulation Studies

There have been seven previous simulation studies that investigated the behavior of the biological CR models that we consider here, five were in a wildlife context, and the remaining two were in a software engineering context. These are reviewed below to elucidate the similarities and differences with our study.

The simulations conducted by Otis et al. [38] used actual values of N in the range of 100 to 800. While these numbers may be appropriate in a biological context, they are larger than the true number of defects that one would expect in an inspected artifact. Although, some authors have noted that such simulations use unrealistically high population sizes for many field studies in the biological sciences, where unrealistically high is defined as $N > 100$ [34]. Therefore, they may not even be appropriate for a biological context. A more realistic value that we use, also used in [49], is 30 defects in a document. Furthermore, the relevant part of this simulation study focused on the accuracy of the total population size estimate, whereas we are concerned with the reinspection decision accuracy.

Chao [15][16] evaluated her Mh model estimator using a Monte Carlo simulation where it was compared to the Jackknife. This simulation used population sizes ranging from 200 to 400, and 5, 7 and 10 capture occasions, and very low capture probabilities. As noted above, the typical N values in software engineering would be expected to be smaller, and the studies do not indicate performance with two captures (inspectors). Another simulation by Chao to evaluate her model Mt estimator used N values of 500 and 1000 with 40 occasions [17]. Finally, a larger simulation, also by Chao et al. [18], to evaluate the performance of the Model Mth estimator used N values of 100, 200, and 400. As would be expected, none of the above simulations considered decision accuracy as a means of evaluating the performance of the estimators.

The fifth simulation was performed in [50] , and was performed in a software engineering context. This focused on only two models, which are a subset of the models that we consider in our simulation. The authors also focus on evaluating the accuracy of the prediction of the number of remaining (undiscovered) defects rather than on decision accuracy. Furthermore, the authors assumed five inspectors in their simulation, while we focus on two inspectors. Finally, these simulations assumed artifacts with 100 defects. For the reasons cited above, we consider artifacts with only 30 defects.

The sixth simulation was reported in [49]. The objective of this was to evaluate the suitability of CR models when one is using Perspective-Based Reading (PBR) techniques. We do not focus on PBR in the current study, and assume a Checklist-Based Reading approach. The rationale is based on the results shown in a recent literature review, whereby the authors conclude that CBR is the predominant reading technique in industry [32].

The simulation study of Menkens and Anderson [34] is the most similar to ours, although they were not concerned with decision accuracy. The focus of that study was the evaluation of CR models in studies with small-mammal populations, which are usually small and their capture probabilities are less than 0.30. They used values of N ranging from 50 to 100, and capture occasions of 5, 7, and 10. For the Chapman estimator (model MtCh in our study) they pooled the observations from the different capture occasions into two occasions, a situation very similar to ours. For the MtCh model, they found that it

generally underestimates and did not perform well when the data met the assumptions of model Mh *and* when there was extreme heterogeneity, and/or when the capture probabilities were low. However, when the data met the assumptions of model Mt and capture probabilities were not extremely low, then its negative bias decreased considerably.

4 Research Method

In this section we specify the study points for our simulation, and describe how the different models were evaluated.

4.1 Study Points

For all of our simulations we set the population size to 30 defects. As noted earlier, this is a more realistic value for a population size in a software engineering document. Three sets of variables were manipulated during the simulations: the distribution of defect difficulty, the probability of a defect being found, and the inspector capability.

As was done in a previous software engineering simulation [50], we define two classes of defects: those that are difficult to detect, defects of type A, and those that are easy to detect, defects of Type B. We varied the distribution of the 30 defects into one of these two classes as follows:

- $\{n_A = 0, n_B = 30\}$: all defects are of the easy type
- $\{n_A = 10, n_B = 20\}$: two thirds of the defects are of the easy type
- $\{n_A = 20, n_B = 10\}$: one third of the defects are of the easy type
- $\{n_A = 30, n_B = 0\}$: all defects are of the difficult type

where n_A is the number of defects in class A, and n_B is the number of defects in class B.

The second variable that was manipulated was the probability of a defect being detected. For each of the two classes of defects that are mentioned above we define these as P_A and P_B respectively. It is necessary that $P_A < P_B$. We therefore define the following two possibilities as follows:

- $\{P_A = 0.1, P_B = 0.9\}$: extreme difference in detection probabilities
- $\{P_A = 0.4, P_B = 0.6\}$: moderate difference in detection probabilities

The third variable that we manipulate is the general defect detection effectiveness of the two inspectors themselves (i.e., their ability to detect defects), which we denote as P_X and P_Y for inspector X and Y respectively. These were defined as follows:

- $\{P_X = 0.1, P_Y = 0.9\}$: one inspector is much better than the other in defect detection
- $\{P_X = 0.25, P_Y = 0.75\}$: one inspector is moderately better than the other in defect detection
- $\{P_X = 0.4, P_Y = 0.6\}$: one inspector is marginally better than the other in defect detection
- $\{P_X = 0.3, P_Y = 0.3\}$: both inspectors have the same low ability to detect defects
- $\{P_X = 0.8, P_Y = 0.8\}$: both inspectors have the same high ability to detect defects
- $\{P_X = 0.5, P_Y = 0.5\}$: both inspectors have the same average ability to detect defects

By combining the possible values on these three variables, we end up with 48 study points. Table 4 gives a complete specification of the 48 study points. Note that the study point numbers provided in this table are used later when presenting the results. Also, in Table 4 we specify the probability model that is assumed by each study point. For example, for study point (1) all defects are easy therefore there is no variation in defect difficulty, but the inspectors vary in their capability. Therefore, this is an Mt study point. The CV value in the table is the coefficient of variation [18] which gives an indication of the extent of variation in the probability of detecting a defect (i.e., heterogeneity). The larger the value of CV the greater the heterogeneity.

For each study point 1000 inspections were simulated.

	Assumed Model	CV	Probability of both inspectors finding the same defect (overlap)	Probability of finding a unique defect by the inspection team.
(1) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mt	0	0.0729	0.8271
(2) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mth	0.595	0.0489	0.5844
(3) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mth	1.028	0.0249	0.3417
(4) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mt	0	0.0009	0.0991
(5) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mt	0	0.0324	0.5676
(6) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mth	0.176	0.0264	0.5069
(7) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mth	0.202	0.0204	0.4462
(8) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.1, P_Y = 0.9\}$	Mt	0	0.0144	0.3856
(9) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mt	0	0.1518	0.7481

(10) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mth	0.595	0.1018	0.5314
(11) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mth	1.028	0.0518	0.3147
(12) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mt	0	0.0018	0.0981
(13) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mt	0	0.0675	0.5325
(14) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mth	0.176	0.055	0.4783
(15) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mth	0.202	0.0425	0.4241
(16) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.25, P_Y = 0.75\}$	Mt	0	0.03	0.37
(17) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mt	0	0.1944	0.7056
(18) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mth	0.595	0.1304	0.5029
(19) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mth	1.028	0.0664	0.3002

(20) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mt	0	0.0024	0.0976
(21) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mt	0	0.0864	0.5136
(22) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mth	0.176	0.0704	0.4629
(23) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mth	0.202	0.0544	0.4122
(24) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.4, P_Y = 0.6\}$	Mt	0	0.0384	0.3616
(25) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.3, P_Y = 0.3\}$	MO	0	0.0729	0.4671
(26) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.3, P_Y = 0.3\}$	Mh	0.595	0.0489	0.3311
(27) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.3, P_Y = 0.3\}$	Mh	1.028	0.0249	0.1951
(28) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.3, P_Y = 0.3\}$	MO	0	0.0009	0.0591
(29) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.3, P_Y = 0.3\}$	MO	0	0.0324	0.3276

(30) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.3, P_Y = 0.3\}$	Mh	0.176	0.0264	0.2936
(31) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.3, P_Y = 0.3\}$	Mh	0.202	0.0204	0.2596
(32) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.3, P_Y = 0.3\}$	MO	0	0.0144	0.2256
(33) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.8, P_Y = 0.8\}$	MO	0	0.5184	0.9216
(34) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.8, P_Y = 0.8\}$	Mh	0.595	0.3477	0.6656
(35) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.8, P_Y = 0.8\}$	Mh	1.028	0.177	0.4096
(36) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.8, P_Y = 0.8\}$	MO	0	0.0064	0.1536
(37) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.8, P_Y = 0.8\}$	MO	0	0.2304	0.7296
(38) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.8, P_Y = 0.8\}$	Mh	0.176	0.1877	0.6656
(39) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.8, P_Y = 0.8\}$	Mh	0.202	0.145	0.6016

(40) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.8, P_Y = 0.8\}$	MO	0	0.1024	0.5376
(41) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.5, P_Y = 0.5\}$	MO	0	0.2025	0.6975
(42) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.5, P_Y = 0.5\}$	Mh	0.595	0.1358	0.4975
(43) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.5, P_Y = 0.5\}$	Mh	1.028	0.0691	0.2975
(44) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.1, P_B = 0.9\}$ $\{P_X = 0.5, P_Y = 0.5\}$	MO	0	0.0025	0.0975
(45) $\{n_A = 0, n_B = 30\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.5, P_Y = 0.5\}$	MO	0	0.09	0.51
(46) $\{n_A = 10, n_B = 20\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.5, P_Y = 0.5\}$	Mh	0.176	0.0733	0.46
(47) $\{n_A = 20, n_B = 10\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.5, P_Y = 0.5\}$	Mh	0.202	0.0566	0.41
(48) $\{n_A = 30, n_B = 0\}$ $\{P_A = 0.4, P_B = 0.6\}$ $\{P_X = 0.5, P_Y = 0.5\}$	MO	0	0.04	0.36

Table 4: Probabilities associated with the 48 study points.

4.2 Evaluation Criteria

4.2.1 Bias, Failures and Dispersion

For all our simulations we first compute the median relative error for each model across all simulated inspections (denoted med(RE)). The med(RE) gives an indication of a model's bias. As noted earlier,

this would allow us to understand the behavior of the CR models and help interpret the results of the decision accuracy evaluations. Furthermore, we compute the number of times a model fails to provide an estimate. This occurs, for example, due to divisions by zero. Finally, we also evaluate the inter-quartile range (IQR) of the relative error. This provides us an indication of the dispersion in the relative error values (i.e., whether the extent of over/underestimates is consistent). For both the med(RE) and the IQR calculations, case wise deletion of missing values was performed. Missing values occurred when an estimator fails.

Initially we interpreted the bias and dispersion results manually by looking for patterns. This is potentially error prone as for each of the med(RE) and IQR results there are 288 values that need to be interpreted and relevant patterns of behavior identified. We therefore constructed regression trees to model the patterns in the results [6]. The unit of observation for this tree construction process is the study point (i.e., n=48). The regression tree is constructed by recursively creating binary partitions of the observations. The splits are selected to minimize deviance, defined as:

$$d = \sum_i (y_i - \bar{y})^2 \qquad \textbf{Eqn. 5}$$

where y_i is either the med(RE) or IQR value and $\bar{y}$ is the mean value. The use of trees has three advantages:

- They can act as a confirmation of our manual search for patterns
- They did indeed identify subtle patterns that were not identified manually (the reason being that trees can take into account complex interactions)
- They provide a convenient way of presenting the interpretation of the results

During the tree construction process we did not perform any automatic pruning. The reason being that we wanted to identify all patterns, and therefore the trees served more of a descriptive intent rather than a predictive one. In a few cases a tree was manually pruned to remove branches that were of no interpretive value (i.e., they conveyed a pattern that was already identified further up the tree or the RE/IQR difference at the terminal nodes was minor). All trees are presented in the results section.

4.2.2 Decision Accuracy

Following on from the discussion in Section 3.2, here we illustrate how the decision accuracy can be evaluated.

CR models are used to make a binary reinspection decision. For controlling inspections, this decision would be based on whether the effectiveness of the inspection is above a specified threshold. The effectiveness threshold is set to ensure a high quality inspection that does indeed detect most detectable defects in the software artifact. Since we do not know the actual effectiveness, we use the CR estimate to calculate the estimated effectiveness.

Let Q_P be the threshold effectiveness set by the organization, then the decision can be stated in terms of the following inequality:

$$Q_P \leq \frac{D}{\hat{N}} \qquad \textbf{Eqn. 6}$$

where $\frac{D}{\hat{N}}$ is the estimated inspection effectiveness. If this inequality is satisfied, then an artifact is passed on to the following phase. If it is not satisfied, then the artifact should be reinspected.

One can define the whole decision for controlling inspection effectiveness across many inspections as follows:

$$\hat{\lambda} = \begin{cases} 1 & , \quad \hat{N} \le \dfrac{D}{Q_P} \\ 0 & , \quad \hat{N} > \dfrac{D}{Q_P} \end{cases}$$ **Eqn. 7**

where $\hat{\lambda}$ is the decision based on the CR model, and is one (pass) if the estimated effectiveness is higher than or equal to a certain threshold, and zero (reinspect) if it is lower than the threshold.

In evaluating decision accuracy, one can compare the decision based on the estimates, $\hat{\lambda}$, with the decision that would be made if the CR model was perfectly accurate (i.e., always made the correct decision), which we will denote as λ:

$$\lambda = \begin{cases} 1 & , \quad N \le \dfrac{D}{Q_P} \\ 0 & , \quad N > \dfrac{D}{Q_P} \end{cases}$$ **Eqn. 8**

The results of an evaluation study over M inspections can be placed in a confusion matrix as shown in Table 5.

		$\hat{\lambda}$ = 0	$\hat{\lambda}$ = 1	
λ	0	m_{11}	m_{12}	M_{1+}
	1	m_{21}	m_{22}	M_{2+}
		M_{+1}	M_{+2}	M

Table 5: Notation for a confusion matrix with the CR model's decision.

The M value is 1000, which is the number of simulation runs. We define the decision accuracy in terms of the proportion of correct decisions that would be made using the estimates:

$$Decision\ Accuracy = DA = \frac{m_{11} + m_{22}}{M}$$ **Eqn. 9**

However, this definition of accuracy does not take into account the improvement due to the use of the CR model estimates. It was noted in Section 2.1 that reinspections are rarely performed in practice. Hence, the "no reinspection" decision can be considered the default one. If this default decision is the correct one say 90% of the time and the use of CR model estimates also results in achieving the correct decision 90% of the time, then using the CR model estimates does not add any value. Thus, even though correct decisions 90% of the time for CR estimates may seem impressive, under the above condition they are simply an overhead. It is therefore also necessary to consider the default decision.

We propose the following definition of *Relative Decision Accuracy* (RDA) that accounts for improvements over the default decision:[9]

$$RDA = DA - A_d$$ **Eqn. 10**

[9] In this equation we do not normalise by the default accuracy because in many cases the default accuracy can be zero. This will occur if the threshold is set very high, and therefore the correct decision is always to reinspect the document. For our purposes, however, this does not change the conclusions that are drawn during the study.

where A_d is the accuracy obtained when using the default decision, which in our case is always pass. More precisely, A_d can be defined with reference to the following confusion matrix:

		Default Decision: 0	Default Decision: 1	
λ	0	0	m_{12}	M_{1*}
	1	0	m_{22}	M_{2*}
		0	M	M

Table 6: Notation for a confusion matrix with the default decision.

and:

$$A_d = \frac{m_{22}}{M} \qquad \textbf{Eqn. 11}$$

The definition in Eqn. 10 indicates how much better a CR model estimate is beyond the default decision making criterion. It is positive if the CR model decision is better, zero if they are the same, and negative if the CR model decision is worse than the default decision.

If, during our simulation, there was an instance of failure of an estimator (for example, this can happen due to a divide by zero) we assign the estimator's decision to be the same as the default decision. This is intended to mimic what would occur in actual practice, and also to ensure that we remain on the conservative side while evaluating the CR models.

Since our study is focused on the applicability of CR models to code inspections, we use two thresholds obtained from an extensive and careful literature review [8]. During that study it was found that the average effectiveness of code inspections in practice was 0.57, and the most likely value was 0.7. We use these two values for Q_P during our study. The lower threshold is intended to ensure "above average" defect detection effectiveness, and the higher threshold is intended to ensure "best in class" effectiveness.

4.2.3 Relationship Between Relative Error and Relative Decision Accuracy

Here we demonstrate through examples that a simplistic consideration of the relationship between the RE and the RDA can cause misleading conclusions about decision accuracy if med(RE) is used as the only evaluative criterion. We conclude that explicit evaluation of RDA provides a more realistic picture of the performance of CR models for making the reinspection decision.

Eqn. 6 can be reformulated as follows:

$$RE \le \frac{D}{N} \cdot \frac{1}{Q_P} - 1 \qquad \textbf{Eqn. 12}$$

If this inequality is satisfied then the decision is to pass the document, otherwise it should be reinspected.

The expected value for $\frac{D}{N}$ is given in the last column of Table 4. For the sake of our examples we will assume that its variance is negligibly small. This assumption simplifies the presentation but does not affect the conclusion.

We can, a priori, determine the expected value for the right hand of Eqn. 12 for both our thresholds. For example, for study point (1) and the threshold of 0.57, the expected value for the right hand side of Eqn. 12 is 0.45. The 0.45 value represents the maximum value of RE in order to pass the document. We can then determine whether a model with a given med(RE) will make the correct or incorrect decision. We illustrate this through an example.

Say that the med(RE) of our CR model is –0.15 and the $\frac{D}{N} \cdot \frac{1}{Q_P} - 1$ value is –0.1. This means that at least 50% of the time the RE value will be equal to or smaller than –0.15, and therefore at least 50% of the time the decision will be to pass. This happens to be the incorrect decision[10] and to also be the default decision. By considering only the med(RE) and the expected value in Table 4 one would be tempted to conclude that this model will perform badly on this study point. The RDA for only these lower 50% of observations happens to be zero.

Now, let us say that for the remaining 50% of the observations, the RE value is always larger than –0.05. In those cases the inequality of Eqn. 12 is not satisfied and the decision would be to reinspect the document, which is the correct decision. The default decision is still to pass the document and is still incorrect. The RDA for all the observations then would be 0.5, which is a respectable value. Therefore, consideration of the total RDA provides a more accurate picture of how well the model is performing for a particular study point, while the use of the med(RE) would have provided a misleading picture in this case.

Extending the example, consider another CR model for the same study point that has the same med(RE), but where the top 50% of the observations have an RE smaller than –0.1. Then the RDA for this model would still be zero. Therefore, even though this model has the same med(RE) as the model above, the decision accuracy conclusion is quite different.

The above examples illustrate that using the med(RE) only to draw conclusions about decision accuracy *may* provide misleading results. The reason is that decision accuracy is affected by the precision (i.e., dispersion) of the RE and not only by its central tendency. It becomes important, then, to also evaluate and compare the performance of CR models using the decision accuracy when the context is making the reinspection decision.

5 Results

We first present the results in terms of the relative error and number of failures. Then we present the dispersion results in terms of the RE IQR. We follow that with a detailed presentation of the RDA analysis results.

5.1 Evaluation of Relative Error

In Table 7 are the median relative error values for each of the six models for each of the 48 study points. Also, the table includes the number of times out of the 1000 runs the model failed to estimate.

Below we first describe three general patterns, followed by the behavior of each model. For each model we also provide the regression tree that was constructed as an aid to understanding the model's behavior. The variables used for the tree construction are explained below. Model M0 has the most complex behavior and therefore its explanation is the most involved.

We explain the notation for the regression trees with reference to Figure 1. The squares are terminal nodes and the circles are non-terminal nodes. On each branch there is condition. If the condition is true then take that branch. Within each node is the mean med(RE) value for all observations within that node. This provides a general indication of the bias for the study points that match the conjunctiuon of conditions leading up to that node. For example, for the top rightmost terminal node we can say that the study points with OVERLAP greater than 0.0104 *and* Tdiff greater than 0.65 have a mean med(RE) value of 0.576. The value below a node is the deviance for the tree up to that node.

---

[10] If $\frac{D}{N} \cdot \frac{1}{Q_P} - 1$ is negative that means that the actual effectiveness is below the threshold, and therefore the correct decision is to reinspect.

	MO		MTMLE		MHJE		MTCH		MHCH		MTHCH	
	Median RE	No Failures	Median RE	No Failures	Median RE	No Failures	Median RE	No Failures	Median RE	No Failures	Median RE	No Failures
1. Mt	1.433	104	-0.133	104	0.166	0	-0.1	0	3.433	104	5.2	89
2. Mth	0.766	216	-0.366	216	-0.2	0	-0.366	0	2.033	216	2.5	234
3. Mth	-0.166	473	-0.633	473	-0.566	0	-0.633	0	0.433	473	0.75	421
4. Mt	-0.866	977	-0.866	977	-0.9	194	-0.9	0	-0.733	977	-0.833	970
5. Mt	0.833	363	-0.3	363	-0.233	0	-0.233	0	2.7	363	3.079	389
6. Mth	0.633	437	-0.366	437	-0.3	0	-0.333	0	2.266	437	2.406	484
7. Mth	0.433	533	-0.466	533	-0.4	0	-0.4	0	1.433	533	1.77	538
8. Mt	0.1	640	-0.516	640	-0.5	0	-0.5	0	1.033	640	1.22	661
9. Mt	0.266	8	-0.1	8	0	0	-0.066	0	0.9	8	0.227	5
10. Mth	-0.1	26	-0.366	26	-0.3	0	-0.333	0	0.433	26	0.027	37
11 Mth	-0.466	173	-0.633	173	-0.6	0	-0.6	0	-0.133	173	-0.466	189
12 Mt	-0.866	948	-0.866	948	-0.9	207	-0.9	0	-0.733	948	-0.9	944
13 Mt	0.233	113	-0.166	113	-0.3	0	-0.166	0	1.033	113	0.282	124
14 Mth	0.1	162	-0.233	162	-0.366	0	-0.2	0	1.033	162	0.313	230
15 Mth	0.033	261	-0.3	261	-0.433	0	-0.3	0	0.666	261	0.137	277
16 Mt	-0.066	389	-0.366	389	-0.516	0	-0.333	0	0.666	389	0.066	392
17 Mt	0.033	4	-0.066	4	-0.1	0	-0.033	0	0.4	4	-0.144	0
18 Mth	-0.266	13	-0.333	13	-0.366	0	-0.316	0	0.033	13	-0.340	12
19 Mth	-0.566	110	-0.6	110	-0.633	0	-0.6	0	-0.3	110	-0.587	107
20 Mt	-0.9	925	-0.9	925	-0.9	186	-0.866	0	-0.833	925	-0.833	938
21 Mt	-0.033	73	-0.1	73	-0.333	0	-0.133	0	0.633	73	-0.066	70
22 Mth	-0.1	109	-0.166	109	-0.4	0	-0.2	0	0.466	109	-0.106	111
23 Mth	-0.166	174	-0.233	174	-0.466	0	-0.233	0	0.433	174	-0.187	189
24 Mt	-0.166	303	-0.3	303	-0.533	0	-0.3	0	0.366	303	-0.194	320
25 MO	-0.1	106	-0.166	106	-0.4	0	-0.166	0	0.45	106	-0.155	109
26 Mh	-0.433	209	-0.466	209	-0.6	0	-0.433	0	0.033	209	-0.4	241
27 Mh	-0.733	449	-0.766	449	-0.766	7	-0.7	0	-0.566	449	-0.737	481
28 MO	-0.933	971	-0.933	971	-0.9	449	-0.933	0	-0.933	971	-0.9	972
29 MO	-0.3	360	-0.333	360	-0.566	0	-0.333	0	0.366	360	-0.3	375
30 Mh	-0.433	435	-0.466	435	-0.633	0	-0.433	0	0.066	435	-0.4	447
31 Mh	-0.466	518	-0.566	518	-0.666	0	-0.5	0	-0.083	518	-0.5	556
32 MO	-0.566	641	-0.6	641	-0.733	2	-0.566	0	-0.166	641	-0.533	642
33 MO	-0.033	0	-0.033	0	0.066	0	0	0	0.066	0	-0.358	0
34 Mh	-0.266	0	-0.3	0	-0.233	0	-0.266	0	-0.2	0	-0.507	0
35 Mh	-0.566	2	-0.566	2	-0.533	0	-0.533	0	-0.433	2	-0.656	1
36 MO	-0.766	808	-0.766	808	-0.833	37	-0.733	0	-0.566	808	-0.75	839
37 MO	-0.033	1	-0.033	1	-0.066	0	-0.033	0	0.3	1	-0.222	1
38 Mh	-0.066	4	-0.066	4	-0.166	0	-0.1	0	0.3	4	-0.24	2
39 Mh	-0.066	10	-0.1	10	-0.233	0	-0.1	0	0.4	10	-0.212	14
40 MO	-0.066	34	-0.1	34	-0.3	0	-0.133	0	0.466	34	-0.155	43
41 MO	-0.033	2	-0.033	2	-0.1	0	-0.033	0	0.316	2	-0.212	0
42 Mh	-0.3	10	-0.333	10	-0.366	0	-0.333	0	0	10	-0.406	11
43 Mh	-0.566	103	-0.6	103	-0.633	0	-0.566	0	-0.3	103	-0.635	101
44 MO	-0.866	924	-0.866	924	-0.9	181	-0.85	0	-0.733	924	-0.833	925
45 MO	-0.1	62	-0.1	62	-0.333	0	-0.133	0	0.466	62	-0.155	71
46 Mh	-0.166	103	-0.166	103	-0.4	0	-0.2	0	0.433	103	-0.187	96
47 Mh	-0.166	168	-0.266	168	-0.466	0	-0.233	0	0.366	168	-0.202	165
48 MO	-0.266	282	-0.3	282	-0.533	0	-0.3	0	0.366	282	-0.25	291

Table 7: Median relative error and number of failures for each of the models and study points.

5.1.1 Extent of Overlap

When the inspection team has low capability, then it is expected that the overlap in defect detection between the two inspectors approaches zero quite frequently (i.e., $f_2 \rightarrow 0$). In the decision trees this is exemplified by low values of the variable OVERLAP. In all decision trees the variable OVERLAP was selected, indicating that it is an important one for explaining the behavior of the CR models. In all six trees, whenever there is a split on the OVERLAP variable the lower OVERLAP value branch underestimates. For example, in Figure 1 the split at the root shows that the low OVERLAP branch (left branch) has a much larger underestimation (node value of –0.866) than the high OVERLAP branch (right branch with node value –0.07). This indicates that low OVERLAP leads to underestimation, and can be confirmed by inspecting Table 7.

For study points with the *lowest* probability of inspectors finding defects in common (OVERLAP) all of the models will underestimate considerably (see study points (4), (12), (20), (28), (32), (36), and (44)). Furthermore, models MO, MtMLE, MhJE, MhCh, and MthCh will have large numbers of failures under these conditions, making them clearly unusable when the inspectors find few defects in common.

5.1.2 Extent of Heterogeneity (CV)

We would expect that heterogeneity would have a minimal impact on the RE values of models MhJE, MhCh, and MthCh. However, this is not the case. Models MhCh and MthCh are affected by heterogeneity, whereas MhJE is not. This is evident in the respective decision trees (see Figure 5 and Figure 6). However, for the former two models the impact of CV is secondary to the impact of OVERLAP and inspector capability differences.[11]

Whenever CV is high (high heterogeneity) the tendency is for the RE value to decrease (i.e. tending towards a negative bias). For example, in Figure 1 the split on the CV value of 0.8115 indicates that the lower CV branch (left branch) has a much smaller underestimation than the high CV branch (right branch).

5.1.3 Extent of Inspector Capability Differences

We define the variable "Tdiff" as the difference in the expected defect detection probability between the two inspectors. For example, for stuy point (1) this would be 0.8. We would expect that models MtMLE, MtCh, and MthCh would be minimally impacted by differences in inspector capabilities. However, for model MthCh Tdiff is the most important variable that explains its behavior, indicating a strong sensitivity to inspector capability differences. This can be seen by looking at the top 8 study points in Table 7 and comparing them to the other study points.

Inspection of the decision trees indicates that as inspector capability differences increase, CR models that are affected by this variable tend to have a larger RE (i.e., tending towards a positive bias). For example, in Figure 1 the split on the Tdiff value of 0.65 indicates that the lower Tdiff branch (left branch) has a negative bias, whereas the higher Tdiff branch (right branch) has a positive bias.

5.1.4 Model MO

The regression tree for model M0 is shown in Figure 1. Study points that meet the assumptions of model MO or that depart minimally from them (i.e. zero or low Tdiff and zero or low CV) and where the probability of both inspectors finding a defect (OVERLAP) is relatively high, then MO will estimate relatively accurately (see study points (33), (37), (40), and (41)). For MO study points, model MO tends to fail frequently and has large underestimation with relatively low defect overlap (see study points (25), (29), (45), and (48)).

The behavior of model MO for non-MO study points is affected by the three factors mentioned above as follows:

- As the number of defects found by both inspectors (overlap) decreases, so does the extent of underestimation of this model.

---

[11] This is because the splits on CV occur below the splits on the other variables.

- As the CV increases, so will the extent of underestimation of this model.
- As the differences in the capabilities of the inspectors increase, so does the extent of overestimation of this model.

When there exist combinations of the above (e.g., high CV and low defect overlap) then the model's bias is in the direction predicted above if the combination results in bias in the same direction, or if the combination has biases in different directions, then, in general, it will tend to overestimate. We consider some examples to illustrate the point.

Mh study points with low CVs and relatively high probability of overlap will have a small negative bias (see study points (38) and (39)). When the CVs are low and the probability of overlap is relatively low MO will underestimate (see study points (30), (31), (46) and (47)), that have high CVs and a relatively high probability of overlap, M0 will still underestimate (see study points (34), (35), (42)), or that have a high CV and a relatively low probability of overlap will also underestimate (see study point (26), (27), and (43)). Therefore, any combination of the first two factors leads to underestimation.

For Mt study points the tendency is to overestimate, especially as the difference in inspector capability increases. However, this is balanced by the magnitude of the probability of overlap, which can cause MO to underestimate. For example, study point (24) has a mild difference in inspector capabilities and a low probability of overlap. This leads to underestimation. Study point (5) has a large difference in inspector capability and also a low probability of overlap. This leads to overestimation.

The behavior of model MO under Mth study points is the most dependant on the above three factors. For example, if the difference in inspector capability is low and CV is low, but the probability of overlap is also low then it will underestimate (see study points (22) and (23)). If CV is high and the probability of overlap is also high, then it will also underestimate (see study point (18)). When the differences in inspector capabilities increases and CV decreases, then it overestimates (see study points (6) and (7)).

From this exposition we can see that model MO only works well when its assumptions are met (i.e., no differences in the probability of finding defects and no differences in inspector capabilities) *and* when the probability of defect overlap high. Under other conditions, its exact behavior can vary dramatically depending on the extent of departure from MO assumptions and the extent of defects found by both inspectors. Such sensitivity does not recommend its use unless its assumptions are known to be met.

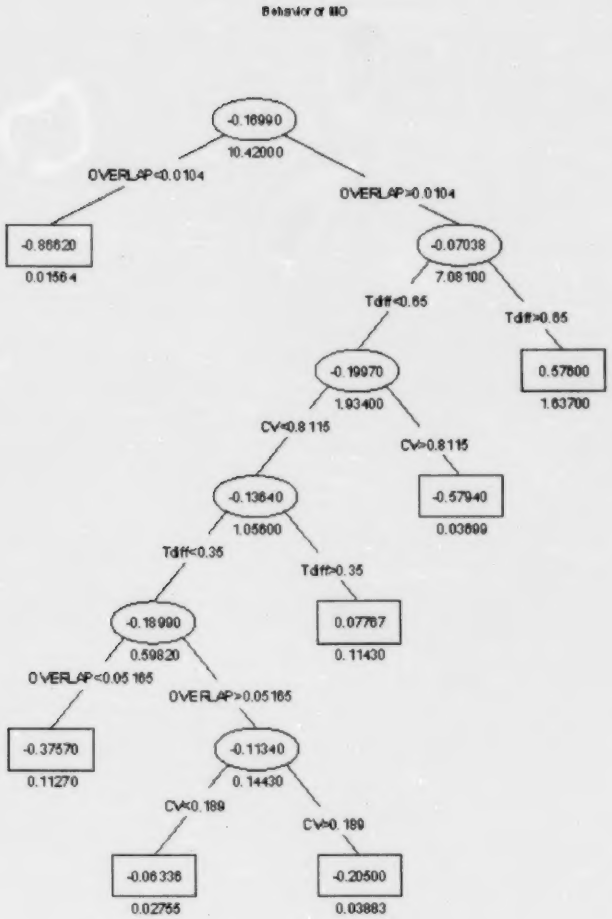

Figure 1: Decision tree explaining the relative error of model M0.

5.1.5 Model MtMLE

The regression tree for model MtMLE is shown in Figure 2. As the OVERLAP decreases, the model MtMLE tends to underestimate considerably, even when its assumptions are met (see study points (1), (5), (8), (9), (13), (16), (21), and (24)). When CV is zero or low and OVERLAP is high then the bias of model MtMLE approaches zero. Therefore, for a subset of MO study points model MtMLE works well

(see study points (33), (37), and (41)). In other situations this model will have a nonnegligible negative bias due to a large CV, even if OVERLAP is large.

Therefore, when the assumptions model MtMLE are violated it underestimates considerably. Even if its assumptions are not violated, if OVERLAP is not sufficiently large it will still underestimate.

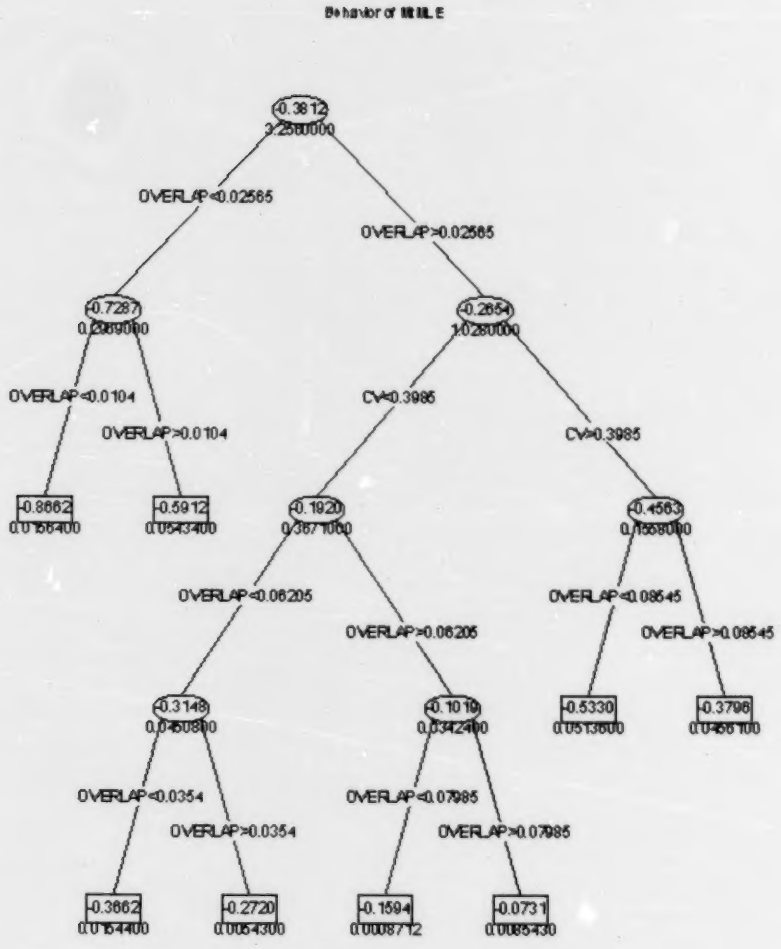

Figure 2: Decision tree explaining the relative error of model MtMLE.

5.1.6 Model MhJE

The regression tree for model MhJE is shown in Figure 3. Similar to other models, the Jackknife estimator exhibits negative bias when OVERLAP is small. This is compensated for when the difference between inspectors increase (as noted earlier, larger Tdiff leads to overestimation). As the OVERLAP increases MhJE performs well for MO and Mt study points (for example see study points (33), (37), (41), (9), and (17)). Otherwise, the Jackknife estimator will in general underestimate. Even for Mh study points with a large OVERLAP, this estimator exhibits large underestimation (see study points (34), (38), and (42)). In general, if CV is not zero it will underestimate. This is surprising as it indicates that this model would work well when its assumptions are violated (Mt study points), but underestimates when its assumptions are met.

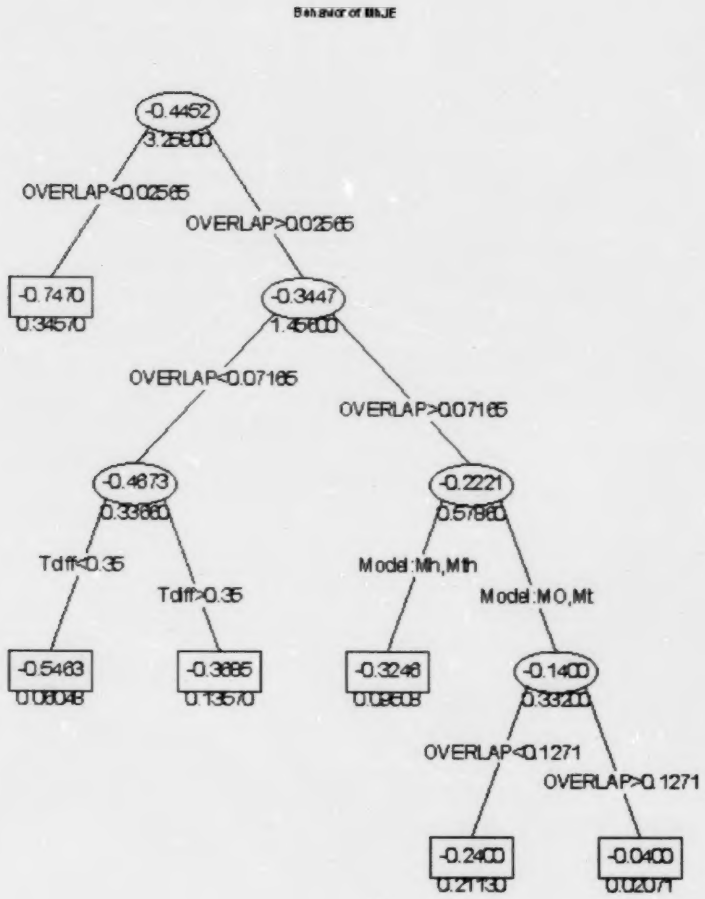

Figure 3: Decision tree explaining the relative error of model MhJE.

5.1.7 Model MtCh

The regression tree for model MtCh is shown in Figure 4. This Chao estimator never fails, which is quite different from all of the estimators considered above[12]. It will, however, generally underestimate defect content. When OVERLAP is low this underestimation increases in general for all types of study points. For MO and Mt study points that have a high OVERLAP, model MtCh does perform reasonably well with its relative error approaching zero (e.g., see study points (9), (17), (33), (37), and (41)). For Mh study points the best performance was obtained when CV was low (study points (38) and (39)), but deteriorated when CV was large and/or when the probability of overlap was low (study points (26), (27), (30), (31), (34), (35), (42), (43), (46), and (47)). These results are consistent with the findings from the simulation in [34].

[12] The reason is that the closed form for this model with two inspectors does not entail a divide by zero when no defects are found in common:

$$\hat{N} = \frac{(n_1 + 1) \times (n_2 + 1)}{(f_2 + 1)} - 1$$

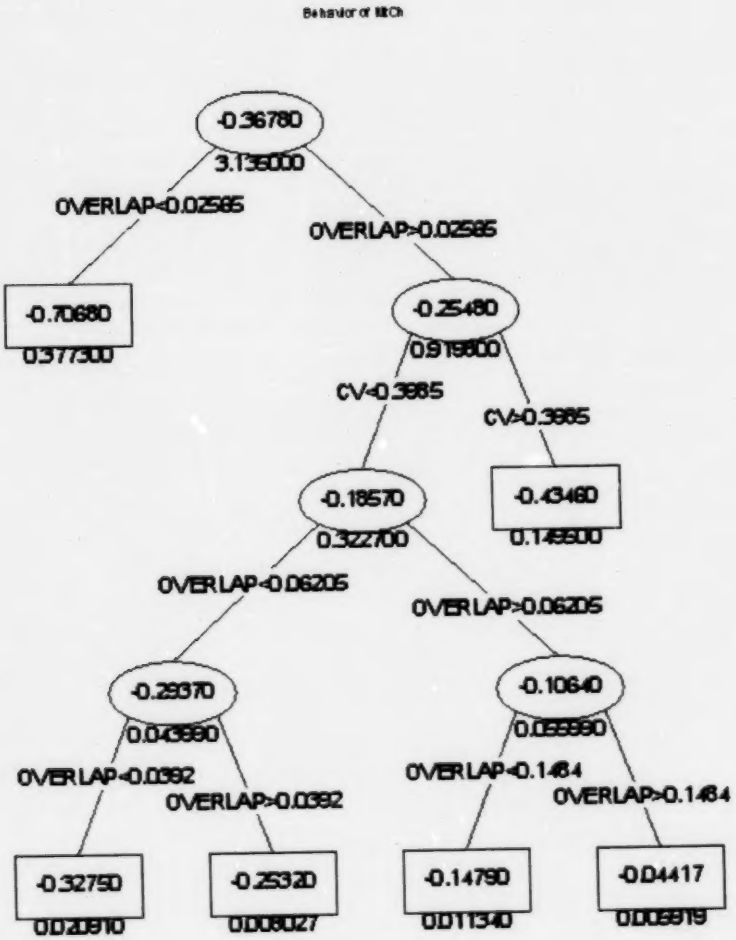

Figure 4: Decision tree explaining the relative error of model MtCh.

5.1.8 Model MhCh

The regression tree for model MhCh is shown in Figure 5. In general, this model exhibits overestimation. Its RE is affected primarily by the differences in inspector capability. Differences in inspector capability violate the assumptions of this model. Therefore, when Tdiff is large the overestimation can be considerable. If the differences are small but the OVERLAP is also small, then MhCh tends towards underestimation. Surprisingly, under the moreorless ideal conditions of a relatively high OVERLAP and a small Tdiff, this model is affected by differences in CV. If heterogeneity is large then this model performs better. If heterogeneity is subtle then it overestimates considerably.

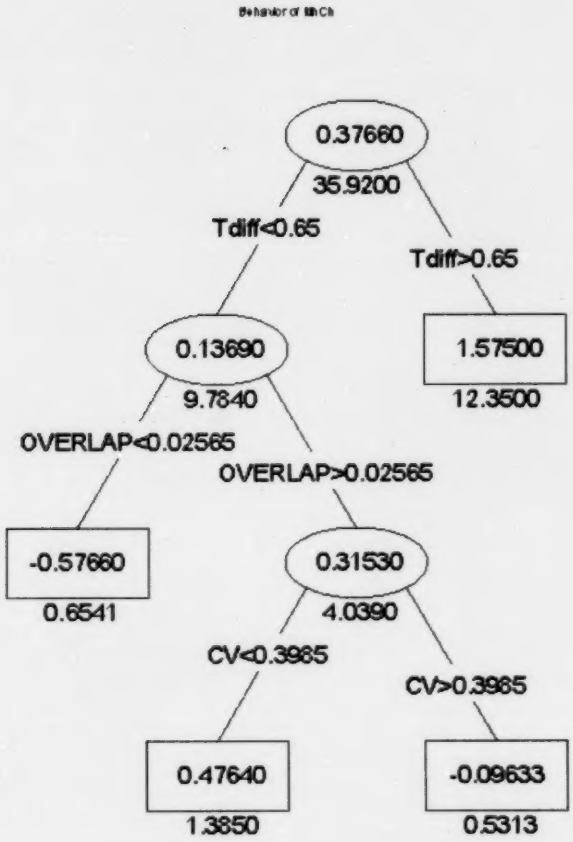

Figure 5: Decision tree explaining the relative error of model MhCh.

5.1.9 Model MthCh

The regression tree for model MtCh is shown in Figure 6. This model has a general tendency for underestimation. Its behavior, however, is consistent with the patterns that we have seen above. It will overestimate if the differences between the capabilities of the two inspectors are large, otherwise its bias will tend towards a negative direction, and eventually as the differences in capability disappear, it will underestimate. For low OVERLAP study points its underestimation will tend to increase.[13]

[13] Note that the lowest right terminal node in the regression tree of Figure 6 may give the impression that for high OVERLAP and Tdiff the RE is small. However, this is not the case as this node combines study points with positive and negative med(RE), and when averaged this gives a value close to zero.

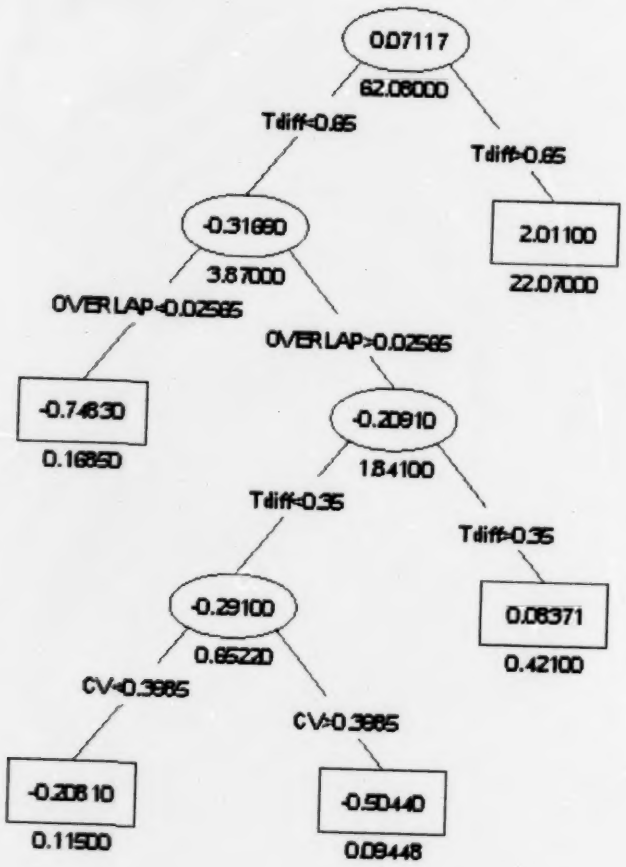

Figure 6: Decision tree explaining the relative error of model MthCh.

5.1.10 Conclusions On Relative Error

By considering the results above on the relative error of CR models with two inspectors, one would be forgiven for concluding that CR models are not usable for two inspectors. Most models tend to exhibit gross over/under estimation, and in some cases fail more than half the time.

We can also compare these results with the results from a previous study that evaluated CR models with two inspectors which used an actual data set [10]. There it was found that all models (except MthCh, which was not evaluated) underestimate. Our results indicate that the CR models sometimes over and sometimes under estimate with two inspectors, depending on a number of other factors. This highlights the importance of performing simulation studies to understand the general behavior of such models.

	MO	MTMLE	MHJE	MTCH	MHCH	MTHCH
1. Mt	1.233	0.2	0.133	0.233	3.525	10.079
2. Mth	1.033	0.166	0.133	0.233	2.7	7.074
3. Mth	0.566	0.1	0.1	0.141	1	1.501
4. Mt	0.133	0.066	0.033	0.1	0.266	0.2
5. Mt	0.8	0.3	0.166	0.466	2.4	5.715
6. Mth	0.8	0.283	0.166	0.4	2.1	4.386
7. Mth	0.833	0.266	0.2	0.433	2.1	3.225
8. Mt	0.7	0.233	0.166	0.4	1.466	2.699
9. Mt	0.466	0.3	0.166	0.3	0.966	0.981
10. Mth	0.4	0.266	0.133	0.233	0.866	0.888
11 Mth	0.333	0.166	0.1	0.2	0.766	0.552
12 Mt	0.041	0.041	0.033	0.166	0.1	0.15
13 Mt	0.866	0.483	0.2	0.433	1.833	1.411
14 Mth	0.8	0.5	0.2	0.433	1.666	1.256
15 Mth	0.7	0.5	0.166	0.408	1.6	1.075
16 Mt	0.666	0.466	0.166	0.433	1.366	1.031
17 Mt	0.3	0.266	0.166	0.266	0.6	0.385
18 Mth	0.3	0.266	0.133	0.266	0.666	0.344
19 Mth	0.233	0.233	0.133	0.2	0.6	0.3
20 Mt	0.133	0.1	0.033	0.166	0.266	0.15
21 Mt	0.633	0.5	0.166	0.433	1.266	0.684
22 Mth	0.566	0.533	0.166	0.433	1.366	0.745
23 Mth	0.666	0.466	0.166	0.466	1.366	0.722
24 Mt	0.6	0.5	0.166	0.466	1.066	0.522
25 MO	0.566	0.533	0.166	0.408	1.366	0.709
26 Mh	0.466	0.366	0.166	0.333	0.766	0.504
27 Mh	0.2	0.2	0.133	0.266	0.366	0.266
28 MO	0.033	0.033	0	0.066	0.1	0.1
29 MO	0.458	0.433	0.166	0.433	0.833	0.570
30 Mh	0.5	0.366	0.133	0.366	0.966	0.45
31 Mh	0.383	0.366	0.166	0.4	0.766	0.447
32 MO	0.266	0.3	0.133	0.333	0.633	0.337
33 MO	0.1	0.1	0.1	0.1	0.166	0.1
34 Mh	0.1	0.1	0.133	0.1	0.166	0.110
35 Mh	0.133	0.133	0.133	0.133	0.233	0.134
36 MO	0.3	0.166	0.1	0.2	0.433	0.197
37 MO	0.3	0.266	0.166	0.266	0.533	0.303
38 Mh	0.3	0.3	0.166	0.266	0.633	0.347
39 Mh	0.366	0.4	0.166	0.3	0.8	0.426
40 MO	0.466	0.466	0.166	0.4	1.033	0.538
41 MO	0.3	0.3	0.166	0.266	0.633	0.330
42 Mh	0.266	0.266	0.166	0.233	0.5	0.292
43 Mh	0.2	0.233	0.133	0.2	0.566	0.286
44 MO	0.166	0.133	0.033	0.166	0.366	0.097
45 MO	0.5	0.5	0.166	0.433	1.033	0.576
46 Mh	0.6	0.5	0.166	0.433	1.166	0.606
47 Mh	0.633	0.5	0.2	0.466	1.3	0.622
48 MO	0.566	0.5	0.166	0.5	1.166	0.535

Table 8: The inter-quartile range for all study points.

5.2 Evaluation of Dispersion

The values of inter-quartile range for all of the models across all study points are shown in Table 8. We would expect that as a model's assumptions are met, its bias will become more consistent and it will have lower IQR. Furthermore, we would expect assumption violations to increase IQR. We would also expect that increases in OVERLAP will lead to reductions in the IQR.

In general we observe that at low or high values of OVERLAP, the dispersion tends to be low. Whereas at moderate values of OVERLAP, the IQR tends to be at its highest. This behavior is consistent across all models.

Another consistent behavior across all models is that smaller values of CV leads to increases in the IQR. For models that attempt to capture heterogeneity (models of type Mh and Mth) this is likely an indicator that these models require large differences in defect detection probabilities in order to produce consistent estimates when there are only two inspectors. However, for the other models this behavior is counter-intuitive because a smaller CV would be closer to their assumptions.

The above points make clear that, even if we select the appropriate model for a particular situation, we may not be ensuring that the bias is consistent.

By consideration of all the regression trees, we can state that Tdiff and OVERLAP are the most important variables in explaining dispersion because they are the ones used for the root node split.

In the following exposition, we will focus on patterns that add to the general ones discussed above.

The regression tree summarizing the results for model M0 is shown in Figure 7. The RE dispersion of this model is most sensitive to the differences in inspector capability. As inspector capability differences increase, this model will have a greater RE dispersion (see study points (1), (2), (3), (5), (6), (7), and (8)). In general as OVERLAP decreases, the dispersion will also decrease (see study points (4), (12), (20), (28)), and also as it reaches high values dispersion will decrease (see study points (33) and (34)). It can be seen that M0 study points do not necessarily have the lowest dispersion, and that reduction in dispersion of these study points is more a consequence of OVERLAP.

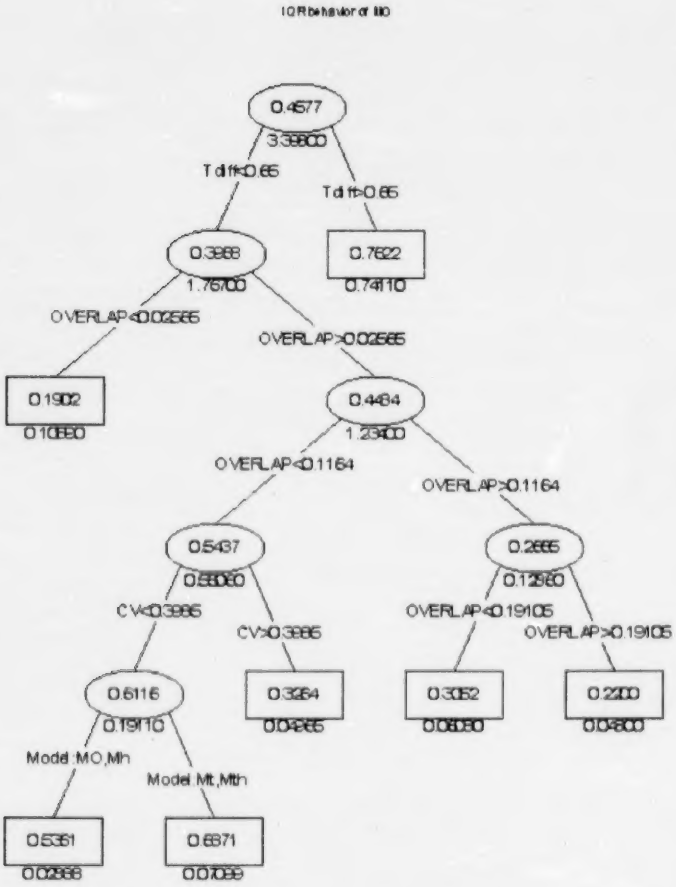

Figure 7: Decision tree showing the IQR behavior of model M0.

The regression tree for model MtMLE is shown in Figure 8. The RE dispersion of this model is affected strongly by OVERLAP. If OVERLAP is not too small (greater than 0.0104) then higher values of CV will tend to have a smaller dispersion. This is counterintuitive as a high CV is a violation of this model's assumptions (for example, compare study points (2) and (3) with study points (6) and (7)).

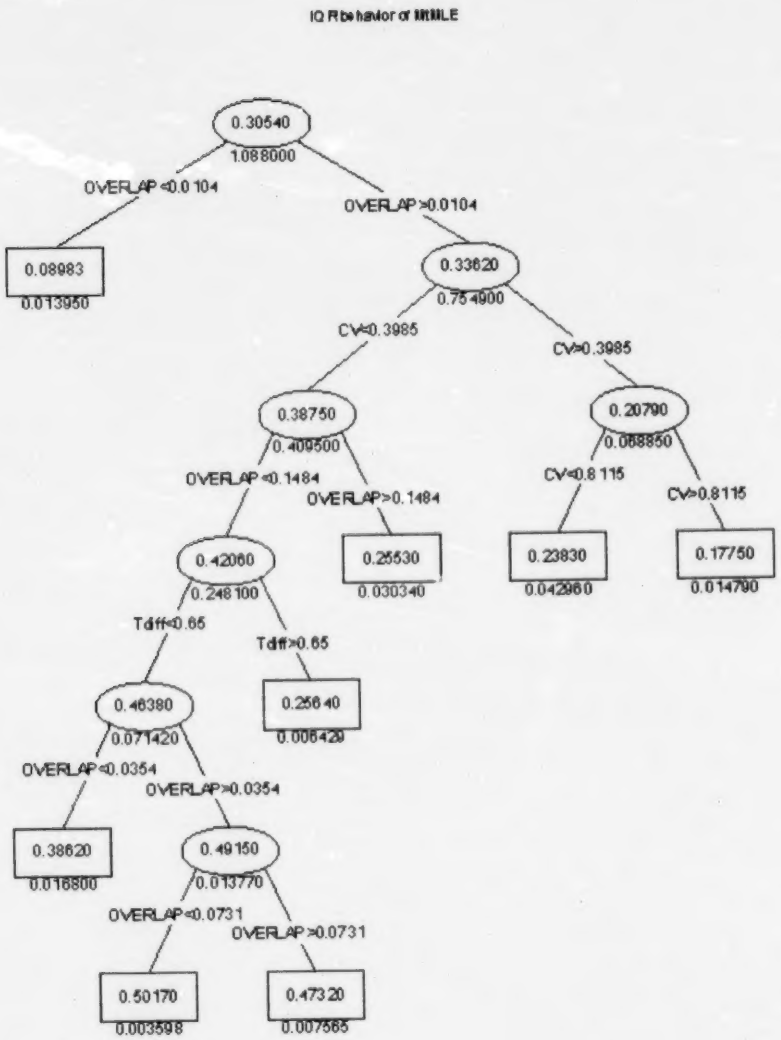

Figure 8: Decision tree showing the IQR behavior of model MtMLE.

The regression tree for model MhJE is shown in Figure 9. In general we can see from Table 8 that model MhJE consistently has the smallest dispersion for all study points. It is interesting to note that when this model's assumptions are violated in the form of Mt study points, the dispersion is not dramatically different from study points that conform to its assumptions. Otherwise, the RE dispersion behavior follows the same general pattern identified above.

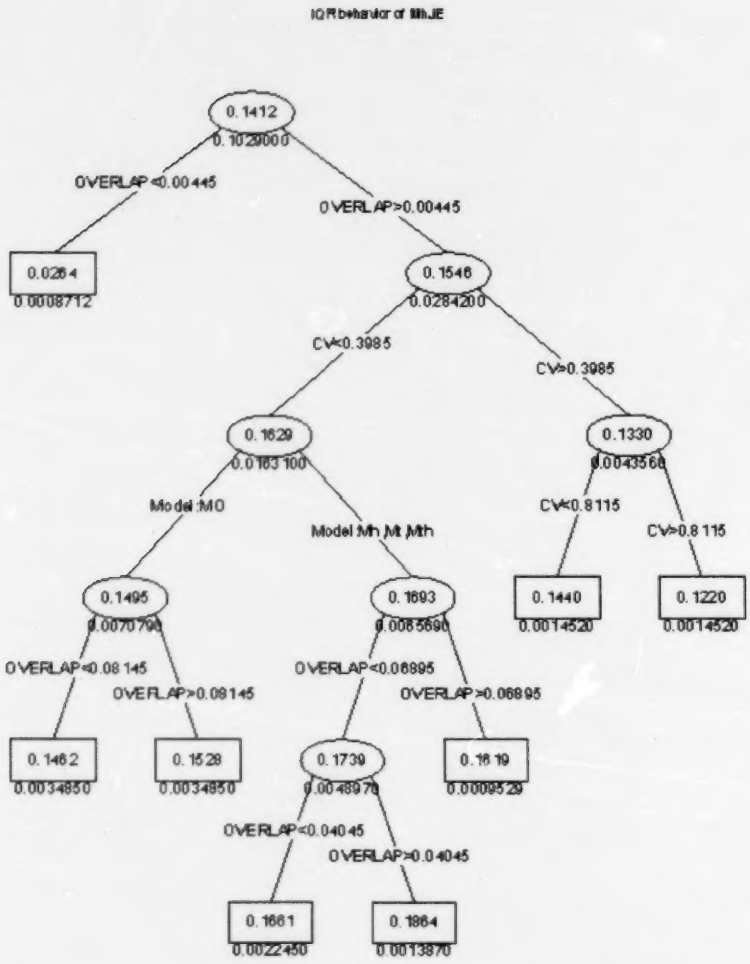

Figure 9: Decision tree showing the IQR behavior of model MhJE.

The regression tree for model MtCh is shown in Figure 10. The RE dispersion behavior follows the same general pattern identified above.

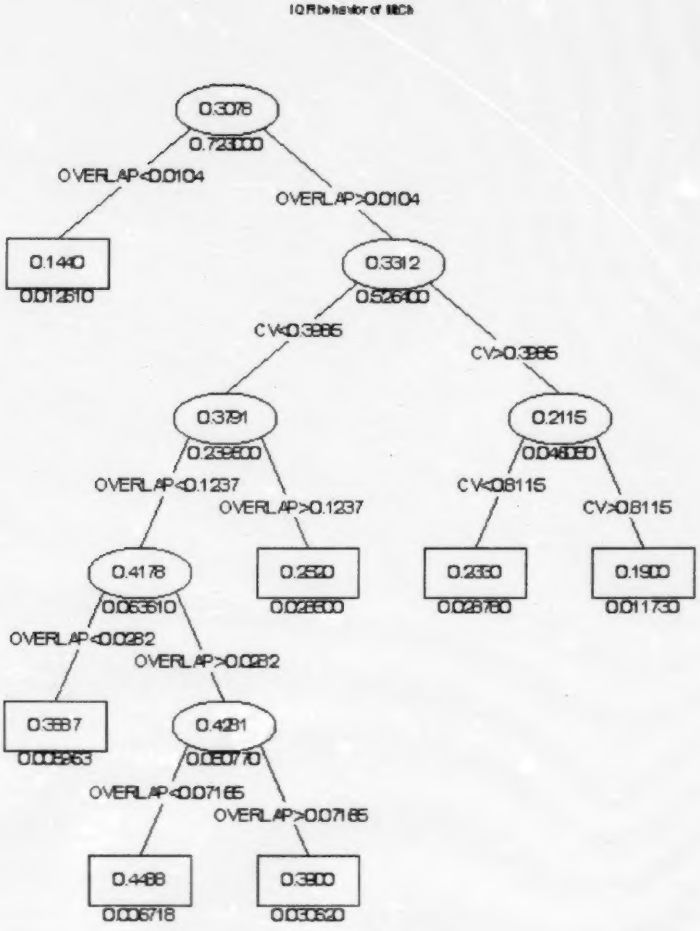

Figure 10: Decision tree showing the IQR behavior of model MtCh.

The regression tree for model MhCh is shown in Figure 11. As would be expected for this model, the greater the differences in inspector capabilities (a violation of its assumptions), the greater the RE dispersion. This is evident by inspecting study points (1), (2), (3), (5), (6), (7), and (8).

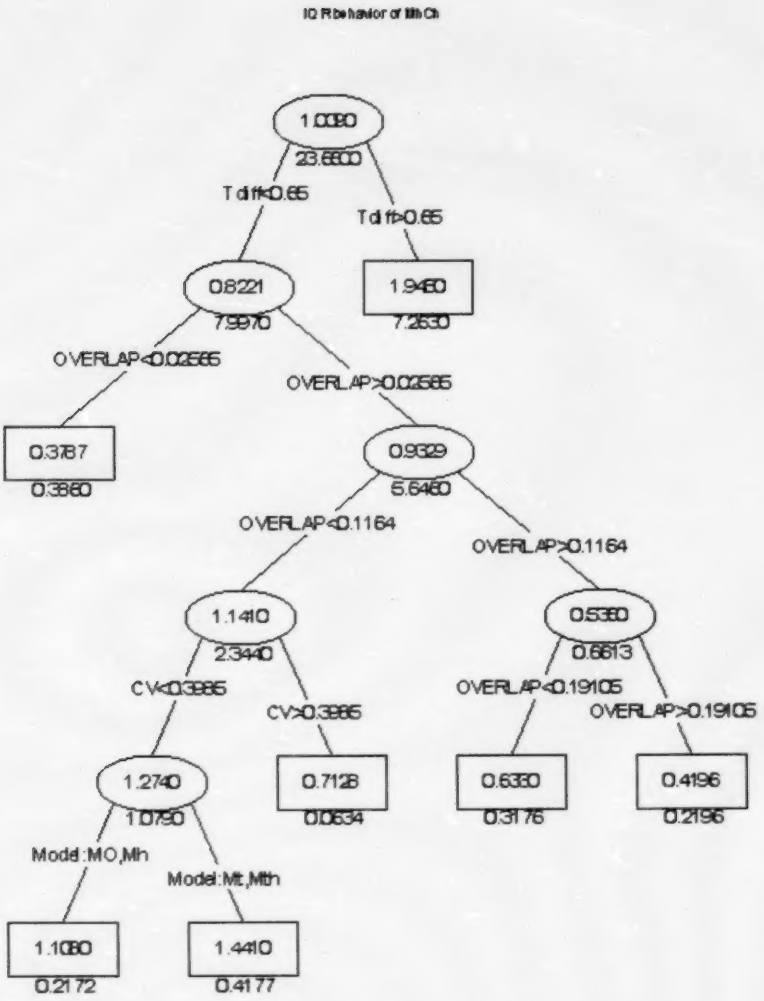

Figure 11: Decision tree showing the IQR behavior of model MhCh.

The regression tree for model MthCh is shown in Figure 12. The RE dispersion of this model is dependent mainly on the differences in inspector capabilities, and increases as this difference increases.

This can be seen from study points (1), (2), (3), (5), (6), (7), and (8). Although not evident in the regression tree, inspection of Table 8 indicates that study points with low OVERLAP tend to also have a low dispersion compared to other study points with similar characteristics (see study points (4), (12), (20), (28), and (44)), and so do study points with a high OVERLAP (see study points (33) and (34)).

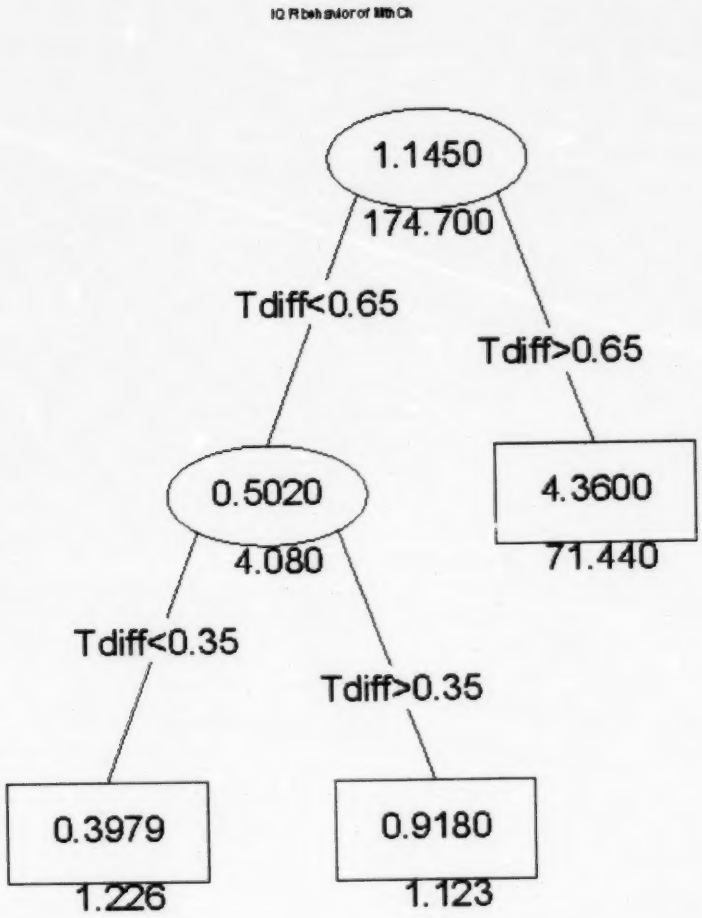

Figure 12: Decision tree showing the IQR behavior of model MthCh.

In general, we can conclude that model MhJE has the lowest dispersion, and that models M0, MtMLE, and MtCh exhibit counterintuitive behavior in that their RE dispersion increases the more their assumptions are met. We have also identified the general patterns for increases and decreases in dispersion.

5.3 Evaluation of Decision Accuracy

The decision accuracy results for both thresholds are provided in Table 9. This includes the DA and RDA results. We only consider models MhJE and MtCh since the other models can have such large failure rates that they cannot be seriously recommended for practical usage with two inspectors, even if their med(RE) and IQR values did exhibit favorable values. For the RDA results, we have bolded the entries in these table that exhibit performance as good as or better than the default decision.

It is convenient to separate the discussion into those study points that have expected effectiveness above the threshold (see the last column in Table 4), and those that have an expected frequency below the threshold.

5.3.1 High Capability Inspection Teams

When the team capture probabilities (see Table 4) are high (i.e., the team is highly effective), then the correct decision is more frequently to pass the document to the next phase. This means that the default decision is correct more frequently. This presents a bigger hurdle for a CR model to overcome in order to provide value beyond the default decision.

For the lower threshold, study points (1), (9), (17), (33), (34), (37), (38), (39), and (41) have expected effectiveness that are above it. Similarly, for the higher threshold study points (1), (9), (17), (33) and (37) have expected effectiveness that are above it. When the CR models are applied to inspections with these characteristics they tend to exhibit performance that is as good as or worst than the default decision. This is exemplified by the RDA values of zero or less.

The reason is that for these study points, CR models that are perfectly accurate (med(RE)=0) or that exhibit underestimation will frequently make the correct decision. Or, if the CR models overestimate slightly then they will still make the correct decision. It will be seen that this is the case for all of the above study points. Since the correct decision is the same as the default decision the RDA will be close to zero.

The DA values for these study points are, however, very large, indicating a good decision accuracy. This is true for both CR models and both thresholds.

5.3.2 Low Capability Inspection Teams

When the inspection team has low capability (below the threshold), then CR models that are accurate (i.e., med(RE)=0)) or that overestimate will frequently make the correct decision (i.e., reinspect). Conversely, if a CR model underestimates then it could also make the correct decision. But this depends on three factors, the extent of underestimation, the dispersion and the CHALLENGE. We define CHALLENGE as the difference between the expected effectiveness and the threshold. If the difference is large then it is a bigger challenge for the inspection team to attain an effectiveness as high as the threshold. If the difference is small then it is a smaller challenge for the inspection team to attain an effectiveness as high as the threshold.

As would be expected, the smaller the underestimation the more likely that the model will make the correct decision. If the dispersion is large, then a larger proportion of the model's underestimates will not be as extreme. Therefore, greater dispersion will in general improve decision accuracy. As the CHALLENGE increases, then underestimation will still lead to the correct decision. We can interpret the DA results in terms of these patterns.

As we saw earlier, when OVERLAP is low model MhJE and MtCh exhibit extreme underestimation, and therefore their decision accuracy will tend to be low. This is exemplified by study points (4), (8), (12), (20), (28), (36), (44). Model MtCh has a larger dispersion than model MhJE, therefore its decision accuracy will tend to be better for these study points. However, these differences are diluted as the underestimation increases. For these study points, model MtCh still performs considerably better than the default decision, as exemplified by the RDA values.

On study points where the dispersion of MtCh is high and the CHALLENGE is large, the decision accuracy tends to increase. For example, for the lower threshold compare study points (16), (24), (26), (29), (30), (31), (32), and (48) with study points (10), (11), (18), (19), (35), (42). In the former both IQR and CHALLENGE were large, whereas for the latter both were low.

5.3.3 Selection of the Appropriate Model

Based on the above discussion, and with the knowledge about bias and dispersion that we gained from the previous results, we can say that the greatest decision accuracy will be gained when:

- Underestimation is not too extreme: avoid small OVERLAP and CV is small
- IQR is large: medium OVERLAP and CV is small
- CHALLENGE is large: setting challenging thresholds

Furthermore, it is clear that model MtCh is a big improvement over MhJE in terms of decision accuracy. Even though the decision accuracy of MtCh is not always very high, it will consistently provide a decision that is better than the default decision. When the above conditions are met model MtCh will differentiate well between inspections that are above/below the thresholds. Therefore, out of all the six CR models the MtCh model is the one recommended for making the reinspection decision with two inspectors.

To attain the above conditions, the two inspectors should not be looking for different defects (e.g., as in perspective-based reading) since this will potentially lead to a small OVERLAP. Furthermore, the inspectors should not have the same specialization in terms of the defects that they look for (otherwise there will be a large OVERLAP). In addition, there should not be great discrepancies in the difficulty of the defects that exist in the document or the distribution of defect difficulty should not be uniform (i.e., CV should not be very large).

	Lower Threshold (0.57)				Higher Threshold (0.7)			
	MhJE		MtCh		MhJE		MtCh	
	DA	RDA	DA	RDA	DA	RDA	DA	RDA
1. Mt	1	0.000	0.928	-.072	0.934	-.060	0.811	-.164
2. Mth	0.532	**0.000**	0.533	**.001**	0.064	**.006**	0.23	**.190**
3. Mth	0	**0.000**	0.167	**.167**	0	**0.000**	0.182	**.184**
4. Mt	0	**0.000**	0.079	**.079**	0.238	**.238**	0.237	**.237**
5. Mt	0.437	**0.000**	0.46	**.023**	0.104	**.012**	0.405	**.367**
6. Mth	0.195	**0.000**	0.368	**.173**	0.032	**.006**	0.407	**.421**
7. Mth	0.068	**0.000**	0.381	**.313**	0.007	**.004**	0.427	**.445**
8. Mt	0.017	**0.000**	0.419	**.402**	0.001	**.001**	0.441	**.451**
9. Mt	0.977	**0.000**	0.902	-.075	0.801	-.002	0.674	-.083
10. Mth	0.248	**0.000**	0.299	**.051**	0.015	**0.000**	0.272	.269
11 Mth	0	**0.000**	0.182	**.182**	0	**0.000**	0.287	**.289**
12 Mt	0	**0.000**	0.159	**.159**	0.206	**.206**	0.451	**.451**
13 Mt	0.296	**0.000**	0.431	**.135**	0.053	**.002**	0.594	**.621**
14 Mth	0.127	**0.000**	0.431	**.304**	0.009	**0.000**	0.63	**.659**
15 Mth	0.034	**0.000**	0.471	**.437**	0.004	**0.000**	0.656	**.674**
16 Mt	0.011	**0.000**	0.537	**.526**	0	**0.000**	0.688	**.695**
17 Mt	0.926	**0.000**	0.832	-.094	0.61	**.001**	0.523	**.008**
18 Mth	0.136	**0.000**	0.233	**.097**	0.009	**0.000**	0.367	**.376**
19 Mth	0	**0.000**	0.199	**.199**	0.001	**.001**	0.402	**.402**
20 Mt	0	**0.000**	0.2	**.200**	0.222	**.222**	0.554	**.554**
21 Mt	0.229	**0.000**	0.435	**.206**	0.031	**0.000**	0.702	**.711**
22 Mth	0.092	**0.000**	0.457	**.365**	0.003	**0.000**	0.734	**.747**
23 Mth	0.033	**0.000**	0.527	**.494**	0.001	**0.000**	0.778	**.790**
24 Mt	0.005	**0.000**	0.607	**.602**	0	**0.000**	0.818	**.823**
25 MO	0.098	**0.000**	0.487	**.389**	0.008	**0.000**	0.769	**.780**
26 Mh	0.002	**0.000**	0.417	**.415**	0	**0.000**	0.692	**.695**
27 Mh	0	**0.000**	0.347	**.347**	0.018	**.018**	0.578	**.578**
28 MO	0	**0.000**	0.05	**.050**	0.272	**.272**	0.343	**.343**
29 MO	0.001	**0.000**	0.62	**.619**	0	**0.000**	0.835	**.837**
30 Mh	0.001	**0.000**	0.622	**.621**	0	**0.000**	0.828	**.830**
31 Mh	0	**0.000**	0.597	**.597**	0	**0.000**	0.809	**.810**
32 MO	0	**0.000**	0.6	**.600**	0.015	**.015**	0.799	**.799**
33 MO	1	**0.000**	1	**0.000**	1	**0.000**	0.997	-.003
34 Mh	0.946	**0.000**	0.945	-.001	0.379	**0.000**	0.38	**.005**
35 Mh	0.003	**0.000**	0.019	**.016**	0	**0.000**	0.064	**.066**
36 MO	0	**0.000**	0.427	**.427**	0.09	**.090**	0.709	**.709**
37 MO	0.957	**0.000**	0.894	-.063	0.72	**0.000**	0.585	-.038
38 Mh	0.835	**0.000**	0.743	-.092	0.419	**0.000**	0.472	**.169**
39 Mh	0.585	**0.000**	0.532	-.053	0.164	-.001	0.51	**.428**
40 MO	0.292	**0.000**	0.431	**.139**	0.059	0.000	0.671	**.649**
41 MO	0.914	**0.000**	0.82	-.094	0.563	**0.000**	0.5	**.055**
42 Mh	0.123	**0.000**	0.224	**.101**	0.007	**0.000**	0.378	**.388**
43 Mh	0	**0.000**	0.2	**.200**	0	**0.000**	0.441	**.442**
44 MO	0	**0.000**	0.189	**.189**	0.223	**.223**	0.567	**.567**
45 MO	0.194	**0.000**	0.434	**.240**	0.03	**.001**	0.72	**.717**
46 Mh	0.093	**0.000**	0.471	**.378**	0.003	**0.000**	0.738	**.745**
47 Mh	0.027	**0.000**	0.505	**.478**	0	**0.000**	0.775	**.779**
48 MO	0.003	**0.000**	0.597	**.594**	0	**0.000**	0.824	**.828**

Table 9: Decision accuracy and relative decision accuracy results for both effectiveness thresholds.

6 Discussion and Conclusions

Capture-recapture models have been proposed as a means for controlling the effectiveness of software inspections, and in general that they can be used to decide when to stop inspections. In this paper we reported on an extensive Monte Carlo simulation that evaluated the accuracy of CR models for two inspectors in the context of code inspections. This study examined in detail the bias in terms of relative error, failure rates, dispersion of relative error, decision accuracy, and relative decision accuracy. For each of these we identified the conditions under which these evaluative measures will increase/decrease. Furthermore, we were able to draw conclusions about which of the models is most usable for making the reinspection decision, what accuracy to be expected from its use in general, and under what conditions it will perform the best.

The model that we found suitable is MtCh. This model accounts for differences in inspector capability but assumes that defects are of the same difficulty. The estimator is that of Chao [17], but was originally suggested by Chapman [14]. Compared to other models, this one did not fail to provide an estimate under any of the conditions we studied, and therefore is generally usable. It will tend to underestimate if the two inspectors find few or no defects in common and if there are large variations in defect difficulty. Its bias will not be adversely affected if there are large differences in inspector capability. If the inspectors find too few or too many defects in common the dispersion of its relative error will tend to decrease, and if the variation in defect difficulty is large its relative error dispersion will tend to decrease. We did not find evidence that differences in inspector capabilities affect its relative error dispersion. If the organization defines a minimal effectiveness threshold for its inspections, then compared to other models, this model will differentiate well between inspections that exceed the threshold and those that are below the threshold, hence making it conducive to deciding when to stop inspections. When an inspection has an effectiveness that is larger than the threshold, then its underestimation is an advantage in that it will make the correct decision almost all the time. If the inspection is below the threshold, then its large relative error dispersion is an advantage in that it will frequently make the correct decision, and this will always be better than the default decision of always passing a document to the next phase. This model will perform the best in terms of making the correct reinspection decision if the inspectors do not find too many or too few defects in common, if there is not a large variation in defect difficulty, and if the organization sets challenging thresholds for itself.

Our conclusions are inconsistent with an earlier study that evaluated CR models with two inspectors using data from an experiment where the accuracy of CR models was evaluated [10]. In that study the authors concluded that capture-recapture models are not usable with two inspectors, whereas we can conclude that model MtCh is a reasonable choice. We attribute this difference to the use of Monte Carlo simulation, which allowed us to study more conditions (whereas in [10] only one condition was examined) and therefore draw more general and stronger conclusions.

While these results are encouraging for the use of capture-recapture models for making the reinspection decision, admittedly, they are not fully satisfying. First, at a conceptual level taking advantage of bias and lack of precision to make the correct reinspection decision seems cumbersome and lacks parsimony. Furthermore, the decision accuracies, while better than the default decision of always passing the document to the next phase, are frequently below the "psychological" threshold of 70% accuracy. In fact, examining the obtained decision accuracies suggests much room for improvement. We therefore strongly encourage further work on improving capture-recapture models for two inspectors, and using model MtCh as the basis. Specifically, two promising avenues are worthy of consideration.

The first avenue is improving the bias and relative error dispersion of model MtCh. One approach that can be pursued is a Bayesian one. A recent study found that subjective estimates by professional inspectors of their personal effectiveness is very accurate (median relative error of zero), and showed how this information can be used to estimate the defect content for an inspection team [26]. Therefore, there is a basis for using subjective estimates in a Bayesian framework.

The second avenue that ought to be pursued is evaluating the probability of the defect content being greater/smaller than a specific threshold value: a hypothesis testing approach. At least, under these circumstances the inspection team can obtain an indication of uncertainty in the decision of reinspection

or otherwise, and a hypothesis testing approach seems more parsimonious with making a binary decision.

Finally, we also encourage the evaluation of decision accuracy in future studies of CR models since this provides greater insight into the utility of capture-recapture models for making the reinspection decision.

7 Acknowledgements

The authors wish to thank Anatole Kark for his comments on an earlier version of this paper.

8 References

[1] M. Ardissone, M. Spolverini, and M. Valentini: "Statistical Decision Support Method for In-Process Inspections". In *Proceedings of the 4th International Conference on Achieving Quality in Software*, pages 135-143, 1998.

[2] J. Barnard and A. Price: "Managing Code Inspection Information". In *IEEE Software*, 11:59-69, Mar. 1994.

[3] V. Basili: "Evolving and Packaging Reading Technologies". In *Journal of Systems and Software*, 38(1), July 1997.

[4] S. Basin: "Estimation of Software Error Rates via Capture-Recapture Sampling". Technical Report, Science Applications Inc., 1972.

[5] S. Basu and N. Ebrahimi: "Estimating the Number of Undetected Errors: Bayesian Model Selection". In *Proceedings of the International Symposium on Software Reliability Engineering*, pages 22-31, 1998.

[6] L. Breiman, J. Friedman, R. Olshen, and C. Stone: *Classification and Regression Trees*. Wadsworth and Brooks/Cole, 1984.

[7] D. Bisant and J. R. Lyle: "A Two-Person Inspection Method to Improve Programming Productivity". In *IEEE Transactions on Software Engineering*, 15:1294-1304, Oct. 1989.

[8] L. Briand, K. El Emam, O. Laitenberger, and T. Fussbroich: "Using Simulation to Build Inspection Efficiency Benchmarks for Development Projects". In *Proceedings of the 20th International Conference on Software Engineering, pages* 340-349, 1998.

[9] L. Briand, K. El Emam, and B. Freimut: "A Comparison and Integration of Capture-Recapture Models and the Detection Profile Method". In *Proceedings of the Ninth International Symposium on Software Reliability Engineering*, pages 32-41, 1998.

[10] L. Briand, K. El Emam, B. Freimut, and O. Laitenberger: "Quantitative Evaluation of Capture Recapture Models to Control Software Inspections". In *Proceedings of the Eighth International Symposium on Software Reliability Engineering*, pages 234-244, 1997.

[11] L. Briand, B. Freimut, O. Laitenberger, G. Ruhe, and B. Klein: "Quality Assurance Technologies for the EURO Conversion – Industrial Experience at Allianz Life Assurance". In *Proceedings of Quality Week Europe*, 1998.

[12] L. Briand, K. El Emam, B. Freimut, and O. Laitenberger: "A Comprehensive Evaluation of Capture-Recapture Models for Estimating Software Defect Content". Submitted for publication, 1998.

[13] K.. Burnham and W. Overton: "Estimation of the Size of a Closed Population when Capture Probabilities Vary Among Animals". In *Biometrika*, 65:625–633, 1978.

[14] D. Chapman: "Some Properties of the Hypergeometric Distribution with Applications to Zoological Sample Censuses". In *University of California Publications on Statistics*, 1:131-160, 1951.

[15] A. Chao: "Estimating the Population Size for Capture-Recapture Data with Unequal Catchability". In *Biometrics*, 43:783-791, 1987.

[16] A. Chao: "Estimating Animal Abundance with Capture Frequency Data". In *Journal of Wildlife Management*, 52(2):295-300, 1988.

[17] A. Chao: "Estimating Population Size for Sparse Data in Capture-Recapture Experiments". In *Biometrics*, 45:427-438, 1989.

[18] A. Chao, S. Lee, and S. Jeng: "Estimation of Population Size for Capture-Recapture Data When Capture Probabilities Vary by Time and Individual Animal". In *Biometrics*, 48:201-216, 1992.

[19] B. Cheng and R. Jeffery: "Comparing Inspection Strategies for Software Requirements Specifications". In *Proceedings of the 1996 Australian Software Engineering Conference*, pages 203–211, 1996.

[20] J. Darroch: "The Multiple Recapture Census: I. Estimation of a Closed Population". In *Biometrika*, 45:336-351, 1958.

[21] J. Duran and J. Wiorkowski: "Capture-Recapture Sampling for Estimating Software Error Content". In *IEEE Transactions on Software Engineering*, 7(1):147-148, 1981.

[22] N. Ebrahimi: "On the Statistical Analysis of the Number of Errors Remaining in a Software Design After Inspection". In *IEEE Transactions on Software Engineering*, 23(8):529-532, 1997.

[23] S. Eick, C. Loader, M. Long, L. Votta, and S. Vander Weil: "Investigating the Application of Capture-Recapture Techniques to Requirements and Design Reviews". In *Proceedings of the SEL Software Engineering Workshop*, pages 97-102, 1991.

[24] S. Eick, C. Loader, M. Long, L. Votta, and S. Vander Weil: "Estimating Software Fault Content Before Coding". In *Proceedings of the 14th International Conference on Software Engineering*, pages 59-65, 1992.

[25] S. Eick, C. Loader, S. Vander Weil , and L. Votta: "How Many Errors Remain in a Software Design After Inspection ?". In *Proceedings of the 25th Symposium on the Interface*, Interface Foundation of North America, pages 195-202, 1993.

[26] K. El Emam and O. Laitenberger: "An Evaluation of Subjective Estimates of Effectiveness for Controlling Software Inspections". *Submitted for Publication.*

[27] M. E. Fagan: "Design and Code Inspections to Reduce Errors in Program Development". In *IBM Systems Journal*, 15(3):182-211, 1976.

[28] P. Fowler: "In-Process Inspections of Workproducts at ATT". In *AT&T Technical Journal*, 65:102-112, March 1986.

[29] R. Glass: "Inspections – Some Surprising Findings". In *Communications of the ACM*, 42(4):17-19, 1999.

[30] S. Isoda: "A Criticism on the Capture and Recapture Method for Software Reliability Assurance". In *Journal of Systems and Software*, 43:3-10, 1998.

[31] S. Kusumoto, A. Chimura, T. Kikuno, K. Matsumoto, and Y. Mohri: "A Promising Approach to Two-Person Software Review in an Educational Environment". In *Journal of Systems and Software*, 40:115-123, 1998.

[32] O. Laitenberger and J.M. DeBaud: "An Encompassing Life-Cycle Centric Survey of Software Inspection"., International Software Engineering Research Network (ISERN) Technical Report ISERN-98-14, Fraunhofer Institute for Experimental Software Engineering, 1997. To appear in the *Journal of Systems and Software*, 2000.

[33] O. Laitenberger, K. El Emam, T. Harbich: "An Internally Replicated Quasi-Experimental Comparison of Checklist and Perspective-based Reading of Code Documents". International Software Engineering Research Network, Technical Report ISERN-99-01, 1999.

[34] G. Menkens and S. Anderson: "Estimation of Small-Mammal Population Size". In *Ecology*, 69(6):1952-1959, 1988.

[35] J. Miller: "Estimating the Number of Remaining Defects After Inspection". International Software Engineering Research Network, Technical Report ISERN-98-24, 1998.

[36] H. Mills: "On the Statistical Validation of Computer Programs". Technical Report FSC-72-6015, IBM Federal Systems Division, 1972.

[37] M. Ohba: "Software Quality = Test Accuracy X Test Coverage". In *Proceedings of the 6th International Conference on Software Engineering*, pages 287-293, 1982.

[38] D. Otis, K. Burnham, G. White, and D. Anderson: "Statistical Inference from Capture Data on Closed Animal Populations". In *Wildlife Monographs*, 62:1-135, 1978.

[39] H. Petersson and C. Wohlin: "Evaluation of Using Capture-Recapture Methods on Software Review Data". In *Proceedings of the Conference on Empirical Assessment in Software Engineering*, 1999.

[40] K. Pollock: "Modeling Capture, Recapture, and Removal Statistics for Estimation of Demographic Parameters: Past, Present, and Future". In *Journal of the American Statistical Association*, 86(413):225-238, 1991.

[41] A. Porter, L. Votta, and V. Basili: "Comparing Detection Methods for Software Requirements Inspections: A Replicated Experiment". In *IEEE Transactions on Software Engineering*, 21(6):563–575, June 1995.

[42] A. Porter, H. Siy, C. Toman, and L. Votta: "An Experiment to Assess the Cost-Benefits of Code Inspections in Large Scale Software Development". In *IEEE Transactions on Software Engineering*, 23:329-346, June 1997.
[43] E. Rexstadt and K. Burnham: *User's Guide for Interactive Program CAPTURE*. Colorado Cooperative Fish and Wildlife Research Unit, 1991.
[44] P. Runeson and C. Wohlin: "An Experimental Evaluation of an Experience-Based Capture-Recapture Method in Software Code Inspections". In *Empirical Software Engineering*, 3:381-406, 1998.
[45] G. Seber: "A Review of Estimating Animal Abundance". In *Biometrics*, 42:267-292, 1986.
[46] G. Shirey: "How Inspections Fail". In *Proceedings of the Ninth International Conference on Testing Computer Software*, pp. 151-159, 1992.
[47] S. Strauss and R. Ebenau: *Software Inspection Process*. McGraw Hill, 1994.
[48] T. Thelin and P. Runeson: "Robust Estimations of Fault Content with Capture-Recapture and Detection Profile Estimators". In *Proceedings of the Conference on Empirical Assessment in Software Engineering*, 1999.
[49] T. Thelin and P. Runeson: "Capture-Recapture Estimations for Perspective-Based Reading – A Simulated Experiment". Submitted for Publication.
[50] S. Vander Weil and L. Votta: "Assessing Software Designs Using Capture-Recapture Methods". In *IEEE Transactions on Software Engineering*, 19(11):1045-1054, 1993.
[51] G. White, D. Anderson, K. Burnham, and D. Otis: *Capture-Recapture and Removal Methods for Sampling Closed Populations*. Technical Report LA-8787-NERP, Los Alamos National Laboratory, 1982.
[52] T. Wickens: *Multiway Contingency Tables Analysis for the Social Sciences*. Lawrence Erlbaum Associates, 1989.
[53] C. Wohlin, P. Runeson, and J. Brantestam: "An Experimental Evaluation of Capture-Recapture in Software Inspections". In *Software Testing, Verification and Reliability*, 5:213-232, 1995.
[54] C. Wohlin and P. Runeson: "Defect Content Estimations from Review Data". In *Proceedings of the 1998 International Conference on Software Engineering*, pages 400-409, 1998.